PRIJS ƒ 7.25/Bfr. 135

Commodore INFO

EXTRA: Amiga-busware Catalogus

ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS

JAARGANG 5, NO. 3, april/mei 1988



LISTINGS
- Ricgraph 128
- Catch-it 128
- Powertoetsen 64
- Super Screen 64
- Schermwissen 64
- Joystick 64
- Digitale klok 64
- Lockin Man 64
- Pesten 16
- Bio ritme Amiga
- Wave Creator Amiga

23 april Computer Info Beurs RAI - Amsterdam

Vaste rubrieken
- Basic Miniatuurtjes
- Machinetaal
- Basic cursus
- Oud van Goudriaan

GeoFILE

Data-opslag in GeoPaint

Hannover Messe '88

C 128 nader belicht

Commodore Info

Verschijnt 8x per jaar
Jaargang 5, no.3, april 1988

Uitgave:
Sala Communications/SAC

Uitgever:
Vic Sharfman

Redactie:
Ir. L. Sala hoofdredacteur
J. Bodzinga adj. hoofdredacteur
J. Boers eindredakteur
drs. M. de Rooij, drs. H. Zoete, H. Smeenk, drs. U. Schuurmans, R. Goudriaan, B. Munniksma, P.C. Broekhuizen.

Redactiesecretariaat:
R. van Zalingen

Strip: Bert Tier
Illustraties: Ben van Mierlo
Advertentie-exploitatie:
Ing. V. Sala, Ing. B Sala, D. van Vlijmen
Weesperstraat 103
1018 VN Amsterdam
tel. 020-273198

Redactie adres:
Postbus 43048
1009 ZA Amsterdam
tel. 020-228871

Listingtelefoon:
(ma: 17.00-21.00) 02155-25162

Abonnementen en administratie:
Postbus 43048
1009 ZA Amsterdam
tel. 020-248006

Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket.

Abonnement:
Voor 8 nummers *f* 47,50 of BFR. 950 per jaar. Betaling op giro 1585491 t.n.v. SAC/Commodore-Info.
Oude nummers kunt U alleen krijgen bij vooruitbetaling van *f* 6,75 op de bovenstaande rekening.
Ook telefonische opgave voor een abonnement is mogelijk. Bel GRATIS 06-02242222 (teleservice), elke dag tot 20.20 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze telefoonnummers zijn alleen bedoeld voor opgave van NIEUWE abonnementen.

Druk:
NDB, Zoeterwoude

Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

ISSN: 0169-3085

Inhoud van dit nummer

Redaktioneel

De Commodore wereld is altijd vol geweest met allerlei geruchten en roddels. Nog was de ene machine maar pas geannonceerd, of de volgende zou al weer klaar staan. Dat was al zo in de tijd van Jack Tramiel, die er een zeker genoegen in schepte altijd een verrassing achter de hand te hebben. Bij iedere persconferentie of interview was er altijd wel een hint, dat er binnenkort weer. ...enz.

In de loop der jaren heb ik ook al heel wat mooie dingen gezien, zoals de hele reeks portables op basis van de 64, de 128, de PC's en al jaren geleden had Jack ook een klein laptopje, dat toe al met een 32 bits processor werkte. Maar we moeten toch wel constateren, dat het grootste deel van al dat nieuws uiteindelijk geen produktiemodel is geworden.

Toch is er, vooral bij onze Oosterburen, de neiging om zeer geregeld met pseudo-nieuws te komen over weer nieuwe Commodore machines. Dat verschijnt dan in de bladen, compleet met fraaie tekeningen en allerlei interessant klinkende details. We hebben daar nooit zo in geloofd, niet omdat er op de tekentafels geen nieuwe versie van de C-64 zouden staan, maar omdat de kans, dat die in produktie gaan, zeer gering is.

Ondertussen heeft Commodore zelf op de Hannover Messe aangekondigd, met een 68030 workstation te komen en ook een 'Transputer' model in ontwikkeling te hebben. Maar meer dan het ontvouwen van plannen en het zoveelste halfwerkende model in de achterkamertjes van de stand blijkt er niet te zijn. En door ervaring wijs geworden, zijn we tamelijk skeptisch. Misschien zien we volgend jaar inderdaad deze machines, tot dat moment geven we u natuurlijk de door Commodore gegeven details graag door, maar loop niet naar de winkel om ze te kopen, daar vindt u voorlopig slechts de vertrouwde 64, 128 (weer volop leverbaar) en de Amiga's.

L.Sala



Luc Sala's Datakolom

Computers in Japan

Vraagt een andere cultuur ook om een andere computer?Op een recente trip in Japan vroeg ik naar het gebruik van computers op scholen daar. Het antwoord was verbazend. Men deed er nog niet veel aan, maar aangezien de buitenlanders altijd vroegen of men geen computers in het onderwijs gebruikte zou het ministerie dat nu eens wat gaan stimuleren.

Met andere woorden, men had zelf geen behoefte gevoeld aan computers, maar aangezien het schijnbaar elders wel zo was, zou men er eens over denken. Wat is dat nu, zijn die nijvere Japanners, die ons overspoelen met hun computerprodukten, geen gebruikers van de eigen digitale medicijn?

Kevin Strehlo, een computerjournalist, schreef een roman over de opkomst van de computerindustrie in Silicon Valley ("Silicon Harvest"). Daarin gaat zijn hoofdpersoon, een geromantiseerde Bill Byron met veel trekjes van Bill Gates, op een gegeven moment geloven, dat de Japanners met de introductie van de microcomputer een groots opgezet complot zijn begonnen om de Amerikaanse industrie op de lange duur uit te hollen. De grote boosdoener zou dan de spreadsheet zijn, het werken met cijfertjes dat de Amerikaanse zakenwereld van binnenuit zou hebben aangetast. Het werken met cijfers in een beperkte opzet zou leiden tot een overwaardering van de resultaten op korte termijn en men zou blind geworden zijn voor de werkelijk belangrijke gegevens en relaties van een onderneming. Op termijn zou dat leiden tot innerlijk verzwakte ondernemingen, die niet meer investeren in activiteiten en funkties, die niet op korte termijn rendement in hun modellen opleveren. In het boek stelt Byron dan allerlei maatregelen voor om het tij te keren, waaronder zo ongeveer een verbod op het gebruik van spreadsheets. Het is een roman, maar ik was onder de indruk van de analyse, die gemaakt werd.
Strehlo heeft met dit amusant boek met een voor de insiders meer dan herkenbare karikatuur van nogal wat Silicon Valley goeroes, toch ook briljant een van de tekortkomingen van de computer weten te treffen. De nadruk op cijfers, op rijtjes en eventueel grafieken is heel leuk, maar verhult vaak de echt belangrijke dingen. Gevoel voor wat er echt gebeurt, is gauw verloren wanneer men alleen maar naar een spreadsheeet zit te turen.

Terug naar Japan, waar ze in dit opzicht wat anders denken bij ons en in de VS. De hele beslissingsstructuur is daar anders, men zal niet snel op basis van cijfers iets concluderen, zonder ook met de mensen te praten en zich in de achtergronden te verdiepen De computer, bij ons primair een cijfermachine, past daar niet zo erg goed bij. Men ziet het meer als een leuk ding om de moeilijke Japanse lettertekens op papier te krijgen, om te tekenen op het scherm of om spelletjes te spelen.
Nu is de computer in Japan ook nog lang niet zo populair als hier. Het gaat om een paar miljoen, gezien de omvang van de Japanse bevolking niet erg indrukwekkend. Men loopt achter op de VS en West Europa. Er zijn wel een 10 miljoen Nintendo Famicon speelcomputertjes verkocht, maar dat zijn meest dingen met insteekmodules en zonder toetsenbord. Toetsenborden zijn natuurlijk al helemaal niet geschikt voor de grote variëteit in de Katakana, Hirakana en vooral kanji karakters, waarin men leest en schrijft. Er is wel een hele generatie tekstsoftware, die een beetje gekunsteld toch het gebruik van een normaal toetsenbord mogelijk maakt voor Japanse karakterprocessing. Dat is echter in een enigszins betaalbare versie pas ongeveer een jaar op de markt. Verder zegt het begrip database de niet erg alfabetisch ingestelde Japanners ook niet zoveel, men houdt niet van erg geregelde en overzichtelijk opgestelde dingen. Adressen, straatnamen en huisnummers bijvoorbeeld zijn soms wel, maar ook vaak niet goed vastgelegd en zoiets als een postcode is de Japanners nog vreemd. De taxichauffeur, gaat meestal eerst vijf minuten zitten turen in per wijk, straat en blok uitvergrote kaarten, en moet het dan twee keer op de hoek vragen.
Alles bij elkaar is de Japanse manier van werken met gegevens duidelijk anders dan bij ons, en hun computers zijn daar ook wel een beetje op aangepast. Misschien is dat ook de reden, dat MSX uiteindelijk niet de grote zegetocht voor de Japanse electronica-industrie geworden is, die men verwacht had. Er zijn er wel een paar miljoen van verkocht, maar met name in de VS is het nooit goed aangeslagen. In Europa is het verhaal bekend, na een leuk succes in een paar landen zitten we nu met een soort MSX-concentraties in een paar landen.

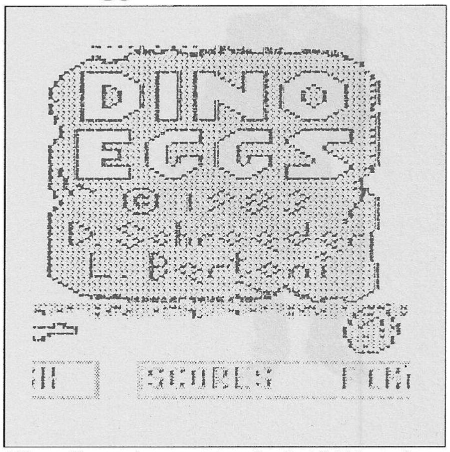
Op dit moment is er op computergebied trouwens een hele nieuwe Japanse invasie op komst, men ziet daar namelijk in Unix een nieuwe kans om hardware te verkopen. Bijvoorbeeld Sony en Sanyo hebben al exportplannen en lieten onlangs hun Unix hardware al in de VS zien. Men mikt met name op de onderwijswereld, daar is Unix heel populair en ziet men grote afzetten opdoemen. Ook hier zouden we kunnen opmerken, dat het multiuser karakter van Unix systemen mogelijk beter past bij culturen, waar het beslissingsproces wat meer hierarchisch is en men niet in persoon samen hoeft te komen om iets te beslissen.

Luc Sala

Te vaak verdwijnen goede spellen voor de Commodore 64 al op de rommelzolder voordat ze werkelijk bekend zijn geworden. Gelukkig snuffelt Rob Goudriaan regelmatig rond om een aantal oudjes nogeens voor het voetlicht te halen

Oud van Goudriaan

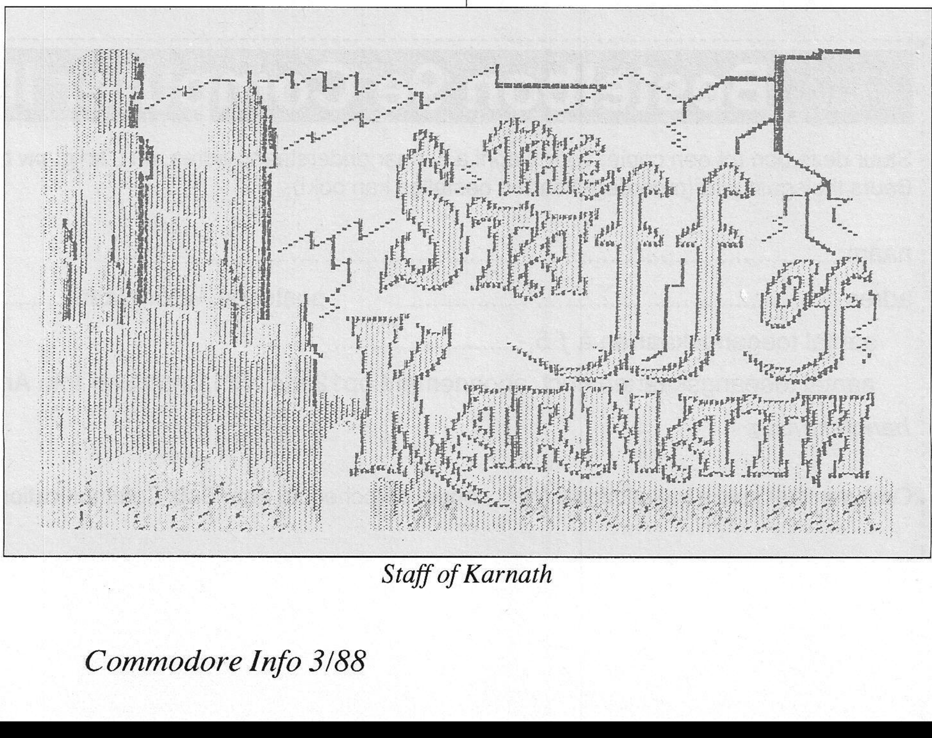
Dino Eggs



Dino Eggs is een spel uit 1983 en behoort dus zeker in deze rubriek thuis. Ditmaal ben je een tijdheer genaamd Tim, en je hebt de nobele taak om jonge dinosaurissen te redden. Dit is te doen door de eieren mee te nemen naar deze moderne tijd. Daarvoor moet je met een tijdmachine naar de oertijd en weer proberen terug te komen met de eieren. Zo gemakkelijk gaat dat natuurlijk niet. Het spel heeft 10 verschillende velden. Reizend in je tijdmachine kan je de eieren transporteren maar je kan maar drie eieren tegelijk vervoeren. Elke keer dat je drie eieren verzameld hebt moet je deze eerst naar de tijdmachine brengen om over te brengen. Uiteraard is de moeder dinosaurus het hier niet mee eens, en zul je hier terdege rekening mee moeten houden. Haar gigantische poten zullen je proberen ervan te weerhouden haar kindertjes mee te nemen. Kom je onder deze poten dan overleef je dit niet. Blijf je opletten, dan is dit wel te voorkomen, door tijdig weg te lopen.
Veilig ben je echter alleen als je een vuurtje stookt, gelukkig ligt er hier en daar wat hout verspreid. Heb je dan een vuurtje gemaakt dan wordt er aangegeven hoelang het vuur nog kan blijven branden, let er op dat je bijtijds hout bij gooit. Behalve moeder Dino zijn er nog meer gevaren, enge slangen, grote griezelige spinnen. Naarmate het spel vorderd komen er steeds meer gevaren bij. Hulp kan je onverwachts krijgen van een bloem, deze plukken en je kan ineens meer dan drie eieren dragen. Kom je een klein dinosaurusje tegen dan moet je deze ook meenenemen, dit lukt alleen door over hem heen te sringen. Vergeet je niet tegen mazelen te laten inenten voor je aan dit spel begint, want daar is alle ellende mee begonnen. Aan het begin van het spel komt er een gigantische rots naar beneden vallen. Achter deze rots komt het verhaal van de dinosaurussen te staan lees dit vooral goed, je kan er je voordeel mee doen.

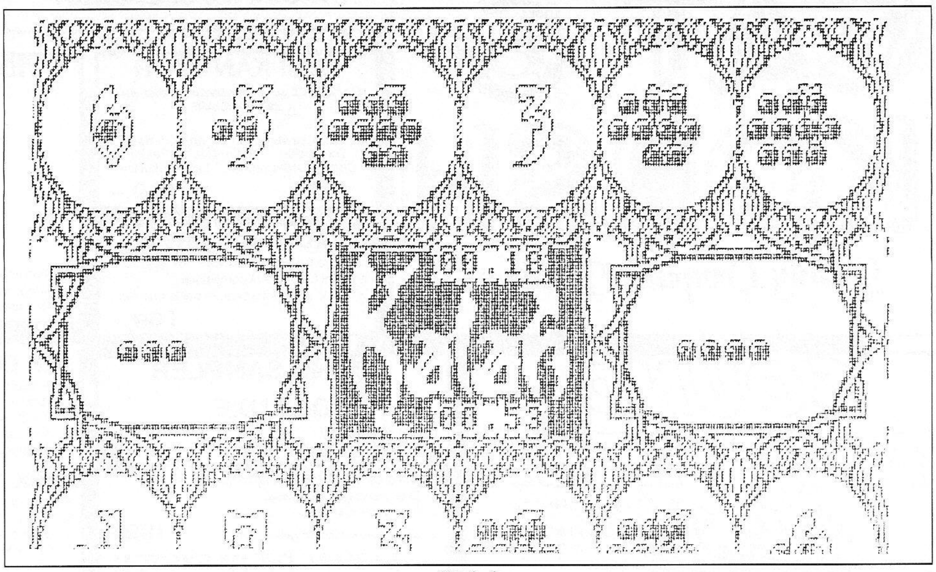
Staff of Karnath

Een sleutel speelt in dit verhaal een sleutelfiguur (hoe kan dit ook anders). Een belangrijke sleutel valt, en blijkt in 16 stukken te zijn gebroken. De brokstukken zijn her en der in een spook kasteel verstopt. Deze stukken worden, als ware het goudstukken op verschillende manieren bewaakt. Er zijn wachters, monsters, heksen, vleermuizen, spinnen en nog veel meer van dat engs. Magische muren, niets blijkt te dol te zijn. Ieder stukje sleutel dat je te pakken krijgt moet naat de obelisk gebracht worden die staat in de kelder van dit slot. Ook deze wordt natuurlijk weer zwaar bewaakt, de botoverde maskers hangen er niet voor niets aan de muur. Zij vuren dodelijke pijlen af zodra je je maar laat zien. Een vogel die hier in de kelder vrij rond vliegt ziet er onschuldig uit, maar pas op voor de eieren die zij laat vallen, ook deze zijn dodelijk. Er wordt om een spreuk gevraagd, blijf je op deze vraag het antwoord schuldig dan kost dit kostbare tijd.Deze spreuken heb je ook nodig om monsters tijdelijk uit te schakelen. Zelf deuren zijn hier soms mee te openen. Deze spreuken zijn opgeborgen in een magische ring die je draagt. Deze is te bedienen door een druk op de vuurknop. Als je een stuk sleutel hebt gevonden dan verschijnt er op het beeld een vierkant, waarin je kan zien welk stuk je bezit. Vergeet niet dit stuk eerst weg te gaan brengen voor je aan het volgende stuk gaat beginnen. Energie is een struikelblok, deze daalt soms heel snel, de enige manier om deze aan te vullen is een sleutelstukje te brengen bij de obelisk. Hierdoor stijgt je energie weer tot 100%. Je beschikt maar over 1 leven dus uitkijken wat je doet. Er zijn ook hele makkelijk kamers bij, hier hoef je alleen een paar wachters te ontwijken, en soms alleen maar af te wachten. In sommige kamers heb je dingen nodig die je eerst uit een andere kamer moet halen. Spinnen zijn allergisch voor een bepaalde spreuk, pak je de goede dan valt het sleuteldeel zo uit het web. In elke kamer is wel wat, vliegende tapijten, vleermuizen, vampiers, trollen eigenlijk te veel om op te noemen. Ieder hoekje, ieder



Staff of Karnath

gaatje zal onderzocht moeten worden, het is dan ook onmogelijk om het spel de eerste keer helemaal uit te spelen. Moet je eerst de moeilijke, of juist eerst de makkelijke kamers nemen? Om dit spel tot een goed einde te brengen zal je op moeten schrijven, wat bij wat hoort. Je zal dit spel verschillende malen moeten spelen voor je hier achter bent. Dit alles mag u zelf proberen. Pas op voor verslaving want daar heb ik zelf behoorlijk last van gehad. Helaas heb ik het spel (nog) niet tot een goed einde kunnen brengen. Mischien lukt het jullie nog wel eens, veel succes, maar je bent gewaarschuwd.



Kalah

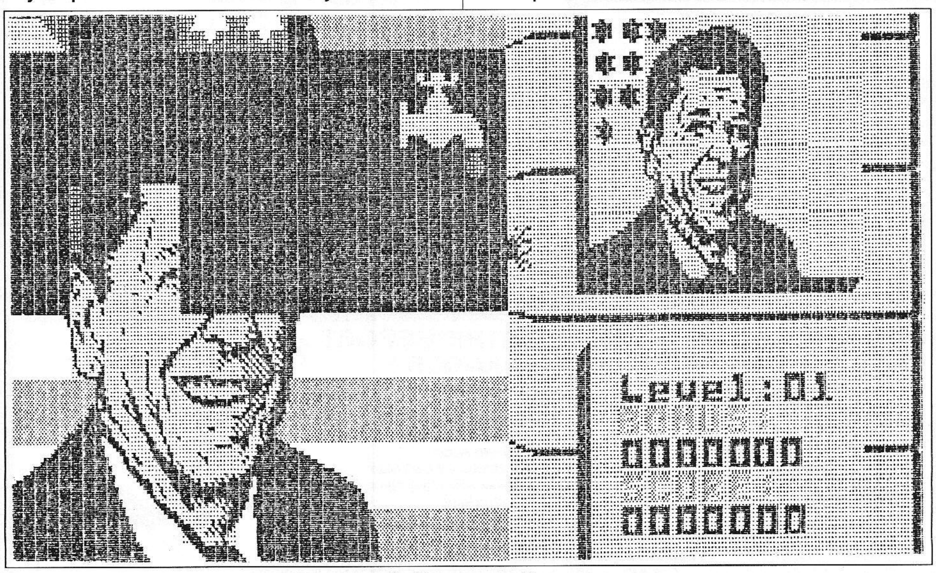
Splitpersonality

Splitpersonality is een uitgebreide schuifpuzzel met afbeeldingen van overbekende wereldburgers. Je begint met de bekendste figuur uit de Amirikaanse politiek, de heer R. Reagan. Door een druk op de vuurknop krijg je een puzzelstukje. Deze moet je proberen op de juiste plaats neer te zetten. Om het geheel wat moeilijker te maken zijn er ook stukjes bij die niets met deze puzzel te maken hebben. Hier moet je goed voor uitkijken, er zijn zeer gevaarlijke dingen bij. Een plaatje met de bom moet je zeer snel weer wegwerken anders hoeft het niet meer en is het spel afgelopen. Hiervoor zijn er in de wanden, zowel boven als beneden, links gaten opgenomen. Helaas gaan deze openingen af en toe dicht, en meestal op het verkeerde moment. Hierin kan je ook de stukjes kwijt die je niet (direkt) nodig hebt. Ook zijn er afbeeldingen van vlaggen, Rusland en Amerika, een vinger dreigend hangend boven een knopje en een pistool, uitkijken dus. Bij bepaalde combinaties zijn er extra punten te verdienen. Een kopje Engelse thee is één van de onderdelen die je tegenkomt bij de afbeelding van de ijzeren dame, M. Tetcher. Let bij dit alles wel op de tijd, want het is een race tegen de klok, ben je niet op tijd klaar dan kost dit je één van je drie kostbare levens. Vele bekende en minder bekende afbeeldingen moet je zo proberen te maken. Hierin ontbreken ook Lady Dy en prins Charles natuurlijk niet. Nieuwsgierig geworden, wie u nog meer tegen komt? Dan zit er niets anders op dan dit spel te gaan spelen.

Kalah

Dit maal eens geen schietspel, race wagens die over het beeld schieten, rare monsters maar een denkspel. Zo op het eerste gezicht lijkt het helemaal niet zo moeilijk om de computer te verslaan, maar je moet toch wel van goude huize komen. Je speelt tegen de computer en hebben beide de beschikking over 7 kuiltjes. Hiervan zijn er zes klein en is er één groot. De grote kuil wordt 'kalah' genoemd, hierin moeten de stenen worden verzameld. Voor we gaan spelen kiezen we eerst het spelnivo. Hierna komt de keusm hoeveel stenen we nemen, en mischien wel de belangrijkste, wie begint er. Degene die begint zegt uit welk kuiltje hij de stenen haalt. Deze stenen worden dan verdeeld, tegen de wijze van de klok in, over de andere kuiltjes. In elk kuiltje moet één steentje worden gelegd. Passeert men een kallah dan wordt ook hierin een steentje gelegd. Wordt het laatste steentje in de eigen kallah gelegd dan mag men nog een keer. Stenen die eenmaal in de kallah liggen mogen er niet meer worden uitgehaald. Wanneer de laatste steen terecht komt in een leeg kuiltje aan de eigen kant van het bord, dan zijn de stenen uit het kuiltje recht tegenover jouw kuiltje voor jou. Deze worden met de steen die ze veroverd heeft in de eigen kallah gelegd. Het spel is afgelopen wanneer er bij een van de twee spelers geen steenjes meer in de kuilen liggen. De steentjes die dan nog in de kuiltjes liggen worden opgeteld bij die in de kalah van de speler. De winnaar is hij, die de meeste stenen in zijn kallah heeft liggen. Ook is er een einde aan het spel gekomen op het moment dat een speler meer dan de helft van de stenen heeft veroverd.

Rob Goudriaan



Splitpersonality

Total loss van de

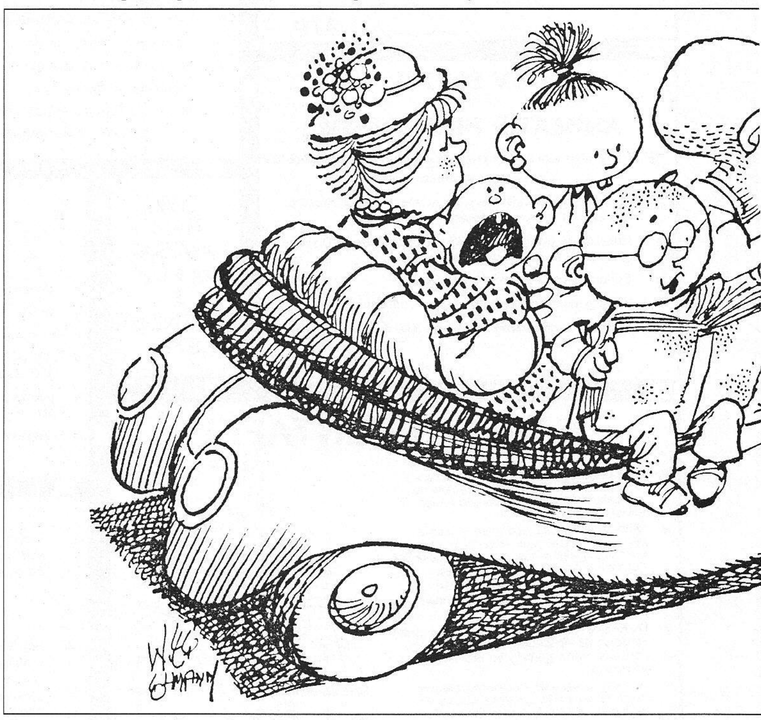
- door Rob Bakker -

"Als de autoindustrie net zo snel was gegaan als de microcomputerindustrie sinds 1981, dan hadden we nu wagens die in 3 seconden van 0 tot 90 kilometer optrokken, de wereld rond konden op een tank benzine en half zo duur waren als zes jaar geleden."
Aldus een deskundige computergoeroe.
Wat ze er niet bij vertellen is, hoe de familie Flopsnee nog een middagje zou kunnen toeren, als de auto-industrie net zo gebruiksonvriendelijk was gebleven als de microcomputerindustrie sinds 1981.

Pa Flopsnee wil gezellig met vrouw en kinderen een rondje gaan rijden in zijn splinternieuwe micro-auto van het merk HAL. Het gezin stapt in de auto en pa Flopsnee probeert het vehikel te starten. Uiteraard gebeurt er niets. Het jongste zoontje Flopsnee, zo'n kiene, etterige whizneuskid, weet raad. "Dan moet je HAL bellen", zegt hij.
Mopperend stapt pa weer uit om de dealer te bellen. Die lui zijn immers niet te beroerd om op hun vrije dag met hun goede pak onder een auto te gaan liggen. Eigenlijk doen ze niets liever en dat is ook de reden dat zo'n man HAL-dealer wordt.
Helaas krijgt pa een automatische telefoonbeantwoorder aan de lijn die voor storingen doorverwijst naar een ander nummer. Na een half uur in-gesprektonen krijgt hij een monteur, die naar zijn klantnummer vraagt.
"Klantnummer?"
"Van uw Service-, Hulp- en Onderhoudscontract."
Pa Flopsnee herinnert zich dat hij die duizend gulden extra voor dat contract verspilling vond en hangt mismoedig op. Gelukkig is hij ook lid van de MCN (Mobiel Club Nederland) en na nog een half uur bellen krijgt hij adviezen om met het goede programma zijn auto op te starten.
Verheugd holt Pa naar buiten en hij krijgt echt de auto gestart. In drie seconden trekt hij op van 0 tot 90 kilometer per uur.
"Kan het niet langzamer?", vraagt Ma Flopsnee benauwd, want 90 kilometer is wat oncomfortabel over een woonerf met drempels.
Pa laat het gas los, maar zo simpel werkt een auto niet. Die is gebouwd om onmiddellijk 90 te rijden en lager, nee, dat zit er niet in.
"O", roept pa, als de volgende drempel met zero wait genomen wordt. "Verlaag de kloksnelheid", roept de akelige whizneuskid van de achterbank.
"De wat?", gilt vader.
De whizneuskid pakt snel een handboek en laat vader allerlei knopjes, handeltjes, palletjes en pieletjes overhalen, waarvan hij niet eens wist dat ze allemaal aan zijn auto zaten. Het resultaat is verbluffend: de auto mindert vaart.

"Alles is mogelijk met deze auto", zegt de whizneuskid. Maar pa denkt met weemoed aan de Trabant die hij ooit had, ook helemaal van plastic en net zo zuinig en snel als dit wonder, maar wel heel erg begrijpelijk in het gebruik.

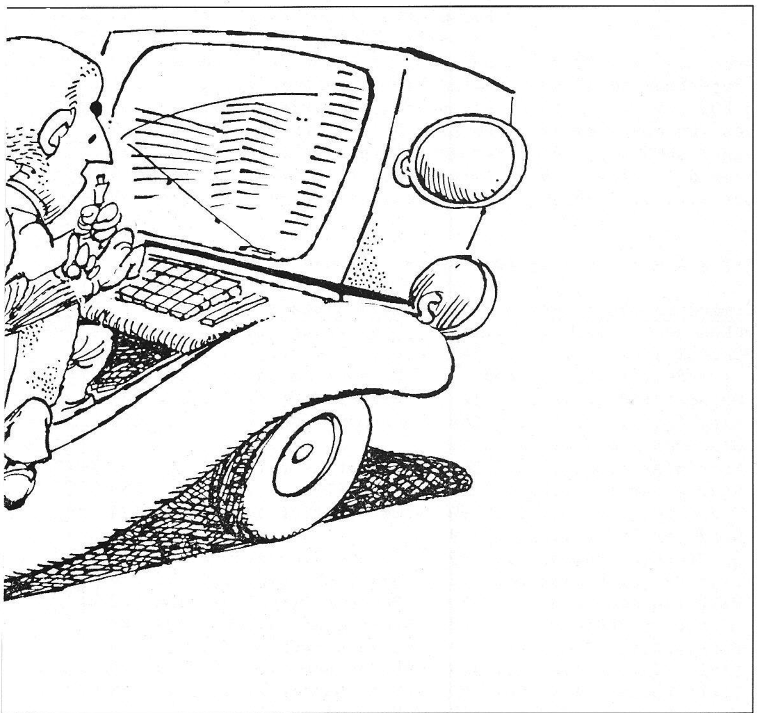
Moeder Flopsnee kijkt trots. Die zoon van ons toch.
In een ontspannen tempo gaat het verder, maar omdat het zondag is, gaat het ook regenen en pa wil de



ruitenwissers aanzetten. Er klinkt een gekras, maar er gebeurt verder niets. Pa probeert het nogmaals, met hetzelfde enge geluid en verder niets. Na enige minuten ziet hij niets meer en moet hij stoppen. Hij verbijt zich als de whizkid uitstapt en vervolgens hoofschuddend naar de ruitenwissers kijkt. "Waar heb je die vandaan, pa?"
"Gewoon, van een goedkoop adresje. De dealer is veel te duur voor een eenvoudige ruitenwisser."
Vader Flopsnee trommelt met zijn vingers op het stuur en zwijgt duister. Gelukkig is hij een zuinig man en had hij de oude wissers bewaard in de kofferbak, voor je kon nooit weten. De wissers worden verwisseld en de tocht wordt voortgezet.
Maar als Pa Flopsnee aan het eind van de straat een bocht wil nemen, dreigt de auto gewoon rechtdoor te gaan.
"Help!", gilt pa en stuurt zich wezenloos.

Microauto

Maar de wielen gaan ieder een verschillende kant op. Op het laatste moment, als een ramp onafwendbaar lijkt, draait pa in paniek de contactsleutel om en de auto staat onmiddellijk stil.
"Je hebt hem gereset", zegt de zoon teleurgesteld. Woedend draait pa zich om. "Ik zal jou eens op je plaats resetten", brult hij. Maar het whizneusjoch verstopt zich achter een paar manuals en daar komt zelfs geen woedende vader doorheen.
"Wat nu?", vraagt moe Flopsnee. "Welke versie van het besturingssysteem zit erin?", vraagt de whizneuskid.



Vader Flopsnee kijkt hem alweer verhit aan. "Gewoon een stuur", roept hij getergd, maar aan de medelijdende glimlach van het joch ziet hij dat de wereld niet zo eenvoudig in elkaar zit.
"Zit het besturingssysteem drie-punt-één er nog in? Of drie-punt-twee? Of al drie-punt-drie?"
Vader kijkt eens naar het stuur. Ziet dat er nu uit als drie-punt-één, of lijkt het meer op drie-punt-twee? Hij krijgt er één-twee-drie een punthoofd van. Hoe zat dat ook alweer met zijn Trabant? Naar links draaien en de auto ging naar links? Hij weet in ieder geval dat hij er nooit problemen mee had.
Inmiddels is de whizkeutelkid in de boeken gedoken en trekt naar voren gebogen allerlei vakjes open in het dashboard, waarvan vader niet eens wist dat ze bestonden.
"Verkeerd besturingssysteem", mompelt het joch. "Ik maak wel een driver."

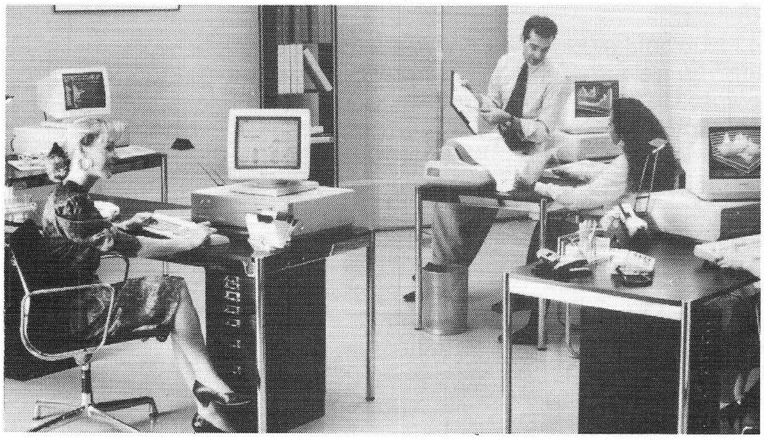
Waarom voelt Pa Flopsnee weer een woedeaanval naderen? Hij was toch de driver? Vroeger, toen een driver nog gewoon automobilist heette, dat waren nog eens tijden. Maar na enig gerommel kondigt zijn zoon aan dat ze verder kunnen. "Maar kijk eerst naar het menu", zegt de whizneuspip. "De auto is nu menugestuurd"."Denk toch niet altijd aan eten", roept vader Flopsnee. "Straks krijg je pas je patatje!" Maar ergens voelt hij dat hij iets verkeerds zegt en grommend start hij de auto. Die trekt in 3 seconden op van 0 tot 90 kilometer per uur. Vader draait het stuur naar rechts en zoals het keurig op het menu staat aangegeven, gaat de auto naar links. In paniek draait pa verder door en de auto gaat geheel menugestuurd achteruit rijden. Pa draait door en nu draait de auto om zijn eigen as en tolt tenslotte met een klap tegen een boom tot stilstand.
"Een crash", zegt de whizneus verdrietig.
Trillend stappen pa en de rest van de familie uit. De schade bestaat uit een deuk in de achterkant. "Misschien wel een nieuw moederbord", taxeert zijn zoon. Ma Flopsnee denkt vetederd dat ze een nieuwe strijkplank krijgt voor moederdag. Pa denkt met weemoed aan de aloude sleepkabel, die vroeger in zijn Trabant lag. Een lusje om de nek van zijn zoon en dan over die tak van de boom....
"In ieder geval moet hij opnieuw geformatteerd worden", onderbreekt de whiznitwit zijn gedachten.
Misschien moet ik hem eens formatteren, denkt vader Flopsnee.
"Maar we kunnen misschien verder op de co-processor", hoort vader de whizkillkid zeggen en hij besluit zich in te houden.
Inderdaad wordt een en ander opgestart en dat maakt vader Flopsnee zo opgelucht, dat hij de opmerking van zijn zoon over een 'open systeem' maar negeert. Wat een smerig taalgebruik heeft die jeugd toch!
Als eindelijk het zondagse ritje achter de rug is, heeft vader Flopsnee een troost; hij heeft nog steeds genoeg benzine in de tank om een keer de wereld rond te kunnen, minus de 15 kilometer die ze vanmiddag hebben afgelegd. Toch wel goedkoop.
"Van die besparingen kan ik bijna de nieuwe onderdelen en het onderhoudscontract betalen", zegt vader Flopsnee optimistisch.
"Vergeet het maar", zegt zijn whizverspilkid. "Die HAL is al vier uur oud en dus overjarig en verouderd en afgeschreven. Niet meer de moeite waard van nieuwe onderdelen. Koop maar een nieuwe!"
Nadat vader Flopsnee met een stevige slash uit de doos van Norton Utilities zijn zoon heeft erased, denkt hij met weemoed terug aan zijn Trabant, die weliswaar twee keer zo duur was als zijn microauto, maar de helft goedkoper.

Rob Bakker is schrijver van boeken en verhalen. Voor Commodore Info schreef hij dit kort verhaal, waarin de zegeningen van de computer op de hak genomen worden.

Traditioneel gebruiken de rivalen Commodore en Atari de Hannover Messe om hun nieuwe plannen en produkten aan te kondigen. Het toneel daarvoor, hotel Maritime in Hannover , is voor beiden gelijk, maar dat komt door de gemeenschappelijke achtergrond: Jack Tramiel is het verbindend element.

Nieuws van de Hannover Messe '88

Voor zowel Commodore als Atari is de Europese markt en dan met name Duitsland, van enorm belang. Commodore heeft in Braunschweig zelfs fabrieken en is daar al jaren marktleider voor de hobbycomputers en belangrijke leverancier op PC gebied.



Op de CeBIT '88 demonstreerde Commodore ook de PC 60 en PC 40 in een netwerkomgeving.

De rivaliteit tussen de beide marktleiders in de huiscomputermarkt is enorm, men heeft weinig goede woorden voor elkaar over en trekt vooral elkaars verkoopcijfers in twijfel. Toch gaat het zowel Commodore als Atari weer goed en beide bedrijven hebben zich min of meer verzoend met een coexistentie als leveranciers van lowend machines met beiden de ambitie om door een fenomenale prijs/prestatie sprong toch in de professionele markt door te dringen.

Atari

Jack Tramiel van Atari kon met enige trots melden, dat men het half miljard dollar peil voor de jaaromzet bereikt had en daarmee in de computergeschiedenis weer een nieuw groeirecord zette. Een deel daarvan is te danken aan de weer groeiende omzet in videospelletjes, die in de VS en Engeland duidelijk bijdragen aan het Atari resultaat.

PC 1,2,3

De reeks MS-DOS modellen van Atari is nu echt op de markt en na de nu beschikbare modellen PC1, PC2, en PC3 zullen later dit jaar ook de PC4 en PC5 uitkomen, die met een 80286 en 80386 processor zullen zijn uitgerust. Voor deze modellen is men nog bezig met een harddisk met uitwisselbare schijven. Er kan op deze uitwisselbare schijven 44 MB aan informatie, en dat alles op één schijf. Men spreekt over een prijs voor alleen de schijf van 120 dollar, die zodra de produktie goed op gang gekomen is moet dalen tot 60 dollar. Daarmee zou men een wel erg goedkoop alternatief voor de harde schijf in handen hebben.

68030 Workstation

Commodore en Atari kijken al jaren, met min of meer gulzige ogen, naar de markt voor professionele workstations. En in het verleden heeft men dan ook weleens aankondigingen gedaan, die op zijn minst gezegd wat voorbarig waren.
We hebben, op dezelfde plaats in Hannover, door de heren Tramiel, Hartman en Shivaz tenminste driemaal een nieuwe Unix machine zien aankondigen, eerst met hun Commodore pet op, nu voor Atari. Steeds ging

Hannover is voor Commodore in Duitsland traditioneel de plek, waar men het nieuws voor het komende jaar voor het eerst laat zien. Ook wereldwijd gezien is dit vaak de plaats, waar men de wereldprimeur van een nieuw systeem laat plaatsvinden. Dit jaar is dat ongetwijfeld de Amiga 2500 AT, die ook voor Unix toepassing is ontworpen.
De specificaties van beide configuraties is:

Amiga 2500 AT

- 68020 Motorola processor (14,2 MHz)
- Harde schijf 40 MB (25 msec)
- AT-bridgecard met 80286 (640 kRAM)
- 3,5 en 5, 25 inch diskdrive (voor zowel PC als Amiga)
- Non Interlace chip-set (640 x 512)
- monitor 1950 (kleuren bi-synchroon)

Amiga 2500 UX

Deze Unix versie van de Amiga 2500 is bedoeld voor de toepassing als workstation en ontwikkelmachine.

- 68020 Motorola processor (14,2 MHz)
- 4 MB (32 bit RAM) intern geheugen
- harddisk van 100 MB (28 msec) - autoboot
- Tape-streamer
- 3,5 inch diskdrive
- Operating System Unix 5.3
- Monitor 2024 (zw/w, 1008 x 1024)

het om een produkt, dat qua prestaties alle concurrenten zou gaan verslaan, tegen een prijs waar niemand tegenop kon. Als dan nu weer een nieuw concept en demomachine wordt gelanceerd, ditmaal met de zeer verse Motorola 68030 processor, maar verder met de zelfde vage beloftes over levering als voorheen, dan kunnen we daar niet echt enthousiast over worden. Atari wil zeker naar de Unix markt, maar is overenthousiast en struikelt steeds over de complexiteit van die markt en de eisen, die daar gesteld worden aan betrouwbaarheid, documentatie en support voor de ontwikkelaars. Men schijnt met AT&T tot overeenstemming gekomen te zijn om de UNIX V versie 3.1 op deze machine te gaan draaien.

Transputer

Zowel Commodore als Atari lieten in Hannover weten, met een model op basis van de Inmos transputer chips te komen. Een transputer is een processor die met hoge snelheid meerdere processen parallel verwerken. Atari mag dat model niet meer ABAQ noemen (er was al een Belgische fabrikant van die naam). Voor beide plannen geldt, dat ze meer in verband met de onderlinge vete tussen de bedrijven moeten worden gezien, dan als produkten voor zelfs de komende jaren. De Commodore transputer is in Braunschweig ontwikkeld en werkt met de T 414 chip op 20 MHZ. Er wordt een verwerkingssnelheid van 10 MIPS genoemd en het besturingssysteem is Helios van Perihelion.

Amiga/Commodore

Op een drukbezochte persbijeenkomst van Commodore was het belangrijkste nieuws het verschijnen van een nieuwe versie van de Amiga, de Amiga 2500 op basis van de 68020 processor, in twee uitvoeringen. De 2500 AT is een AT met als basis 2 MB RAM en een complete nieuwe set chips. Er is een 40 MB harddisk met een toegangssnelheid van 28 ms, waarvan zowel in Amiga-modus als DOS-mode geboot kan worden. Commodore begeeft zich met de 2500 UX op de markt van Unix-computers. Er wordt gewerkt met Unix 5.3 en standaard zijn er 4 MB RAM, 100 MB harddisk, tapestreamer en 3,5 inch drive. Voor deze machine is ook een nieuwe paperwhite monochrome monitor A 2024 met een oplossend vermogen van 1008x1024 punten.

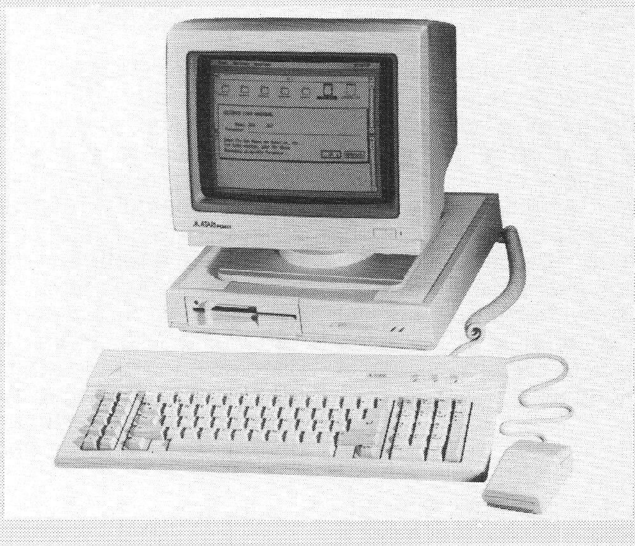
Op de Hannover Messe kondigde Commodore aan, dat men voor de Amiga ook de grotere rekenkracht van de 68020 en 68030 processoren beschikbaar zou gaan maken.

Amiga met sneller hart

Atari/Commodore werkstations

De 68020 kaart

Onder typenummer A 2620 liet Commodore een nieuwe processorkaart zien, waarop een 14 MHZ 68020 draait. Dat is een echte 32/32 bits processor van Motorola, die qua rekenkracht te vergelijken is met de Intel 80386. Met deze kaart draait een Amiga 2000 nog eens vier keer zo snel (400% prestatieverbetering).
De 68020 wordt door het grootste deel van de leveranciers van snelle 'workstations' gebruikt en bedrijven als Sun, Apollo, maar ook Sony maken gebruik van de MIPS van deze chips. De Motorola chip levert namelijk prestaties, die ver boven de Million Instructions Per Second (MIPS) liggen.
In de A 2620 kaart kan men 68881 of 68882 coprocessor gebruiken, die een 14, 20 of 25 MHz kloksnelheid kan hebben, om bijvoorbeeld het vele rekenwerk in CAD/CAM toepassingen nog sneller te maken. Als geheugen zit op de kaart 64 KB ROM met wat extra systeeminformatie en dan 2 tot 4 Megabyte (met dochterkaart) 32 bits geheugen. Dat wil zeggen, dat men het geheugen via een 32 bits bus kan adresseren en dus een veel groter adresbereik heeft, dan bij de 68000, waar dat met een 16 bits bus gebeurt. De besturing van dat 32 bit geheugen gaat met een 68851 chip, die voor 'virtual paging' zorgt.
De A 2620 kaart is compatibel met het AMIGA-DOS operating systeem.



Werkstations

Op de CeBIT/Hannover Messe kondigden zowel Atari als Commodore aan, binnenkort met een workstation voor professionele toepassingen te komen, die gebruik maakt van de Motorola 68030 processor. Dat is een van de krachtigste processorchips, die beschikbaar zijn en zou moeten leiden tot een enorme verbetering in de prijs/prestatieverhoudingen voor werkstations. Dergelijke computers worden gebruikt voor zeer rekenintensieve individuele toepassingen, zoals Computer Aided Design (CAD), software ontwikkeling en DeskTop Publishing. Verder liet Atari weten haar reeks MS-DOS computers nu te kunnen leveren en uit te breiden met machines met 80286 en 80286 processoren. Ook Commodore gaat de reeks PC's uitbreiden en komt ook met meer Amiga modellen, zoals het model 2500.

Amiga 3000

Commodore deed ook een uiterst schimmige aankondiging van een Amiga 3000 met de 68030 processor. Tijdens de presentatie werd een dia getoont met daarop: Commodore 3000; 68030 processor. Verder werd er echter met geen woord over gerept. Ook de 2500 was overigens de eerste beursdag niet op de stand te zien.
Nieuw was de 80386 PC van Commodore, de PC 60, met 3 MB intern geheugen en een 40 of 80 MB harddisk.

Op het gebied van de thuiscomputers had Commodore eigenlijk geen nieuws. Het in het geruchtencircuit verwachtte nieuwe model van de Commodore 64, de 64-3 met ingebouwde diskdrive, lijkt er niet te komen en het gerucht gaat dat Commodore geen geld heeft voor de produktie van het apparaat. Dat is niet zo verwonderlijk gezien de grote activiteit op het gebied van professionele systemen (Amiga 2500 en de PC-lijn). Er is wel al enige tijd een nieuwe 3,5 inch diskdrive voor de C-16, VIC-20, 64 en 128, de 1581. Maar dat was in Hannover al geen nieuws meer. Ook het laatste model in de 128-serie, de 128 DCS met metalen kast, kwam in Hannover niet aan de orde. Er werd wel duidelijk gesteld dat op het gebied van de 128 alleen nog de versie met ingebouwde diskdrive geleverd wordt, en dat er geen nieuwe modellen meer te verwachten zijn.
Voor de Amiga 2000 kondigde men wel een upgrade-kaart met de 68020 processor aan. Deze A 2620 processorkaart heeft twee of vier MB RAM en 64 KB ROM, een 68020 op 14 MHz en coprocessor. Verdere nieuwtjes van Commodore: een kleurenprinter voor de Amiga (zie elders in dit artikel) en een Genlock-board A 2300 voor inbouw in de Amiga 2000.

Ook de producenten van printers hebben de afgelopen maanden niet stil gezeten. De meest interessante machines belichten we hieronder in vogelvlucht.

Printer-nieuws uit Hannover

De tijden, dat de C-64 en C-128 versies van nieuwe printers automatisch werden uitgebracht, zijn helaas voorbij. Star is in dit opzicht een gunstige uitzondering en zelfs Commodore zelf blijkt zich toch voornamelijk op Amiga en PC bezitters te richten met haar nieuwe printer.

Commodore

Commodore presenteerde de MPS 1224 C kleuren-printer. Deze printerserie zal de 2000 en de 2010 machines gaan vervangen, en is speciaal voor de Amiga en AT ontworpen. Deze 24-naalds matrixprinter heeft bij normale printkwaliteit een snelheid van 220 tekens per seconde (bij NLQ 72). De printer is compatible met de Epson LQ-1500 en de IBM 24 XL. De printer kan door een eenvoudig menu worden ingesteld, en heeft zowel een paralelle als een seriële poort. De printbreedte is resp. 136 (Pica) en 163 (Elite).

Canon

Canons nieuwe laserprinters LBP-8IIR en 8IIT hebben twee papierinvoercassette's, zodat er met een druk op de knop van papierformaat gewisseld kan worden. Model R



kan bovendien tweezijdig printen, een doorbraak in het laserprintwerk. Beide modellen kunnen twee cartridges herbergen voor lettertypes e.d.
Canon komt ook met een bubble-jet printer (BJ-130) die een resolutie van 360 dpi kan printen, te vergelijken met de nieuwe HP DeskJet.
De Xerox 3700 is een nieuwe laserprinter die de IBM 3268/87 of 3776/7 nabootst. Er kunnen per minuut maximaal 24 pagina's worden gedrukt, waarmee een capaciteit van 100.000 pagina's per maand gehaald kan worden.
Voor de PC-gebruiker is er een nieuwe kleuren inktjet printer de 4020. Dit model heeft vier inktkleuren waarmee 4.000 kleurnuances bereikt kunnen worden in een maximale resolutie van 240 x 120 punten, waarin over een pagina ongeveer 4 minuten wordt gedaan. De printer kan zowel losse vellen, rollen als folie bedrukken.

Fujitsu

Deze japanse fabrikant kondigde een 'nieuwe generatie' printers aan, waaronder een LED-printer en twee lasermodellen. De RX 7100 LED printer heeft een HP LaserJet + emulatie en een resolutie van 300 dpi voor een prijs van rond de 4.000 mark. Maximale snelheid 5 pagina's per minuut. De RX 7200 en 7300 zijn laserprinters met de meest gebruikelijke emulaties. Ze kunnen resp. 12 en 17 pagina's per minutt afdrukken. Standaard resp. 640 en 2.500 KB intern geheugen en prijzen van 8.000 en 17.000 mark.

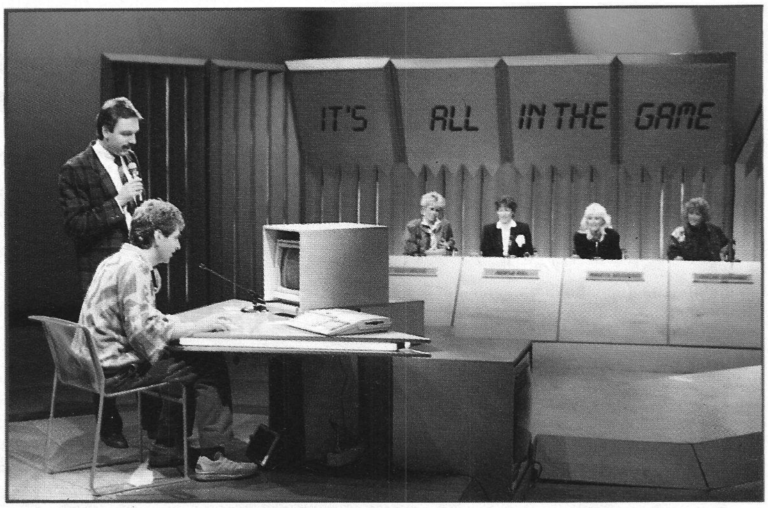
Gestetner

De firma bekend van de stencilmachines liet printers zien met de Canon CX en LBP-20 engines. De GLS800 reeks print 8 ppm, de GLS2000 20 ppm. Er zijn emulaties voor IBM, DEC en Wang computers en voor Diablo, Sprint 5/9, NEC en QMS.



Soft Wir War
Bert tier & Wouter Meijer
HEY!
!
Arcade
SHOP
ZAP
ZAP
VROOAAA
VROAOOOO!
!
!
KLAP!
KLENG
HARGH! HARGH! HARG!
PĀPŪPĀPŪ PĀPŪ PĀPŪ
PĀPŪPĀPŪ
HARG! HARG!
ZO MENEERTJE...DAT WORDT BROMMEN!

Computerspelletjes winnen nog steeds aan populariteit. Dat blijkt ook uit het spelprogramma 'It's all in the Game' van de NCRV. Een uniek samenwerkingsprojekt gaat er nu voor zorgen, dat de TV-games uit dat programma ook thuis kunnen worden gespeeld.



Presentator Henk Mouwe met enkele kandidaten.

IT'S ALL IN THE GAME

De NCRV, René Stokvis Produkties en Sala Communications hebben gemeenschappelijk een uniek projekt opgezet. De computergames, die dit spelprogramma de revue zullen passeren, kunnen op diskette besteld worden, waardoor de kijkers nu ook zelf aktief met de spelletjes 'in de slag' kunnen.

It's All in the Game is vanaf dit seizoen elke maand op maandagavond te zien na het journaal van acht uur. Presentator Henk Mouwe test dan de behendigheid van een achttal kandidaten (in twee teams) op het gebied van computerspelletjes. De formule blijkt uitstekend aan te slaan getuige de kijkcijfers.

Thuis spelen

Om meer mensen de mogeljkheid te bieden de spelletjes te spelen, is nu besloten de games ook op een diskette uit te brengen. In overleg met René Stokvis Produkties (de bedenkers van dit programma) en de NCRV zal Sala Communications (uitgever van diverse computer-magazines, waaronder Commodore Info) de produktie en verkoop van diskettes gaan verzorgen. Zo kunnen ook de thuisblijvers en de fervente computergame-spelers hun eigen kwaliteiten testen met dezelfde spelletjes als in de programma's aan de orde zullen komen.

Diskette I (Alleen voor C-64/128)

Op de eerste diskette staan de volgende spelletjes:

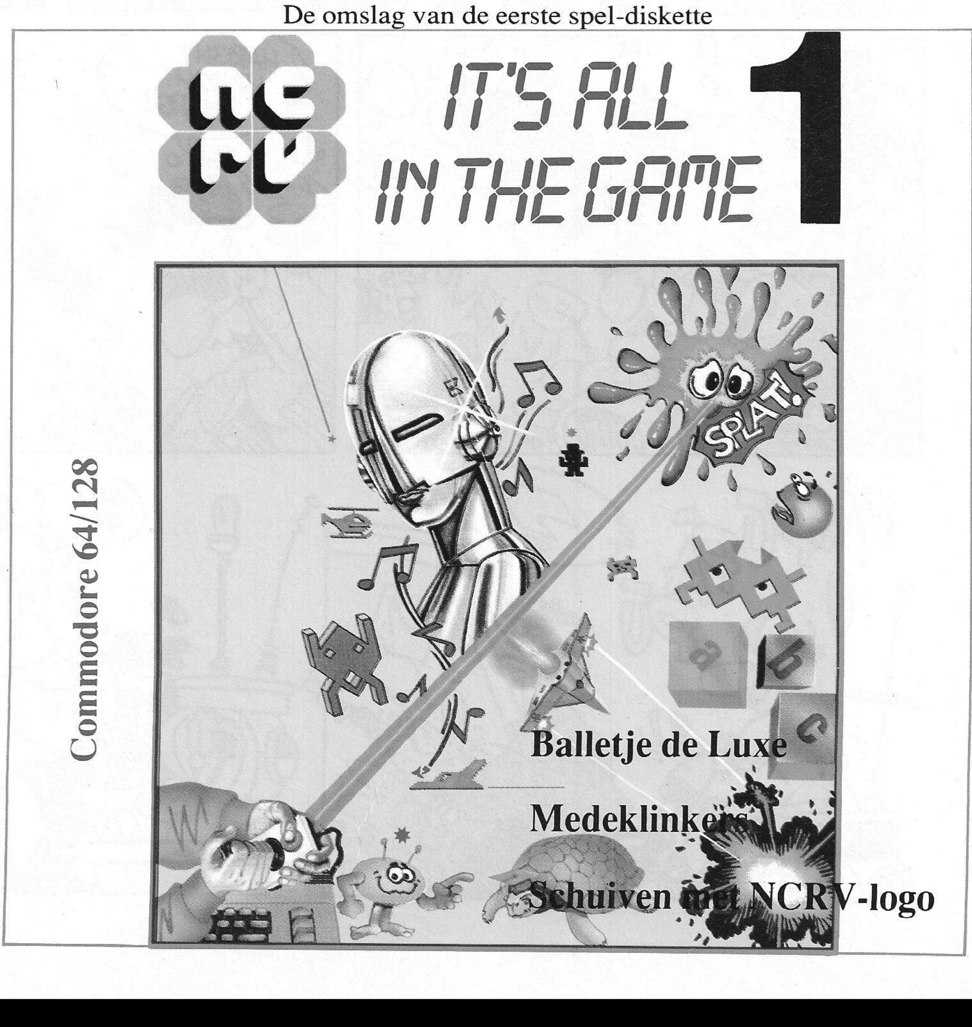
- **Balletje de Luxe**
  één balletje, zes dopjes, en maar raden...
- **Schuif Schuif**
  3 logo's, 8 dopjes. Schuiven in een oogwenk.
- **Medeklinkers**
  16 cotegorieën, vele honderden woorden, maar... géén medeklinkers!

Aanvragen

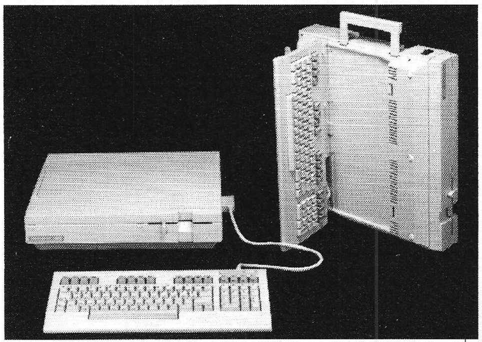
Bestellen van de diskettes met daarop de games kan eenvoudig door ƒ19,95 over te maken op gironummer **1585491** t.v.n. Sala Communications te Amsterdam met vermelding: 'It's all in the game'. U kunt ook een girobetaalkaart opsturen naar Postbus 5570, 1007 AN Amsterdam (ook hierop niet vergeten te zeggen, welke diskette u wilt ontvangen).

Natuurlijk kunt u voor al deze informatie ook naar het programma kijken, dan heeft u dubbel plezier!

De omslag van de eerste spel-diskette



Met ingang van dit nummer zal Commodore Info een aantal pagina's wijden speciaal aan de C128/D. Schrijvers Johan en Johan hopen in deze rubriek hun enthousiasme over de C128 over te brengen op oude en nieuwe gebruikers. Veel plezier.



TRIPLE 128

Wie zijn Johan & Johan?

Johan & Johan zijn twee programmeurs die naast hun normale baan een aantal zeer handige zaken voor de C128/D zullen publiceren. Dit duo heeft al diverse programma's helpen ontwikkelen. Vaak was dit achter de schermen, dus waarschijnlijk zeggen de namen u niets. Wij zullen in het vervolg een aantal pagina's vullen met allemaal handige tips en ontdekkingen omtrent de Commodore 128/D. Wij vragen bij deze de medewerking van alle C128/D gebruikers.

Indeling TRIPLE 128.

De zogenaamde '**Hardware** Freaks' onder ons zullen wij van dienst zijn in een speciale rubriek. Ook zal er elke keer een '**Software**' TEST, van een C128 programma worden gepubliceerd. Deze keer is dat het programma 'FAST 128' van Radarsoft. Vanaf de volgende keer zal er een **Vragen** rubriek komen. Dit speciaal voor onze lezers die gebruik maken van een C128/D. Vergeet dus niet uw vragen en oplossingen naar ons op te sturen! Ja natuurlijk kunt u nog steeds uw eigen tips en truuks naar Commodore INFO insturen. Die tips en truuks mogen ook wel simpel wezen hoor! Als ze maar werken! Werkende tips zullen wij proberen te publiceren. Op deze manier kan een ander er ook van 'snoepen'.

Wat gaan we doen?

Een heleboel verschillende dingen. Maar er zijn nog genoeg personen die bar weinig over de C-128 weten zodat wij op een laag pitje zullen moeten beginnen. Binnen de kortste keren zal dit lage pitje een 'helse brander' worden! Dus alle lezers die gebruik maken van een C128/D zullen niet meer weg te branden zijn van hun Commodore INFO. In de hoogste versnelling, en als in een turbo mode zullen starters beginners worden. In diezelfde mode zullen de gevorderden onder ons, een echte EXPERT worden! Zo zullen wij diverse hard-ware projecten opstarten. Ook zullen wij software tips niet vergeten. Tevens zullen we een start maken met C64/C128 conversie. Met behulp van deze tekst wordt het mogelijk om C64 modem programma's om te zetten voor gebruik op de C128.

De gebruikte lektuur.

Ter ondersteuning bij het werken met de C128 gebruiken wij vele boeken van diverse uitgevers. Natuurlijk allereerst de handleiding van de computer zelf. Verder de INTERN van Databecker, TIPS & TRUCS van dezelfde uitgever, The PROGRAMMER'S REFERENCE GUIDE van Bantam Computer Book, verder hebben wij de beschikking over diverse ROM listings van de C128. Zo ook de SERVICE MANUAL van de C128. Voor de CP/M mode gebruiken wij de USER'S GUIDE, PROGRAMMER'S GUIDE en de SYSTEM GUIDE van DIGITAL RESEARCH. Alle door ons gebruikte boeken kunt u tegen de nodige betaling verkrijgen bij 'Computercollectief'. Het adres is:
Amstel 312 1017 AP Amsterdam.
Voor onze zuiderburen is dat:
Het Computerwinkeltje pvba
M Sabbestraat 39, B-2800 Mechelen.

De gebruikte hardware.

Als hardware hebben wij, de redactie van TRIPLE 128, twee maal een losse C128, een 1570, twee maal een 1571 met snellader en één 1581 disk-drive die gebruik maakt van 3.5 inch diskette's. Als printer gebruiken wij een KX-P1081 van Panasonic en een LX 800 van Epson. Natuurlijk hebben wij een RAM expansion unit (type 1750). Onze redactiecomputers zijn uitgerust met de Power Booster van Master Genius. Hiermee hebben wij een maximale resolutie van 1024*1024 in interlace mode. Met onze C-128 kunnen wij dan ook rustig IFF file's van de AMIGA inlezen. Als monitor maken wij gebruiken van de NEC Multisync en een ANCONA 40/80 monochroom monitor. Tot zover de door ons gebruikte hardware.

Johan & Johan.

VRAGEN

Het insturen van vragen.

Om een antwoord te krijgen op uw vraag zult u het volgende moeten doen. Ten eerste zult u de door u gebruikte hardware moeten omschrijven. Noteer de type nummers en het merk van de door u gebruikte hardware. Vaak kunnen wij niets zonder deze broodnodige informatie. Voor de retourzending dient u te zorgen voor een retourenvelop. Deze envelop zal vergezeld moeten zijn van voldoende postzegels. Alles even op een rijtje. Naam, Adres, Woonplaats (de zogenaamde N.A.W. gegevens voor ons bestand.) postcode, telefoonnummer en uw leeftijd. Dus voor vragen op en/of aanmerkingen kunt u vanaf nu TRIPLE 128 nemen.
Wat voor soort vragen?
Van alles; bijvoorbeeld: Ik krijg het maar niet voor elkaar om een 80 kolommen scherm aan te sluiten op mijn C128. Wat kan ik hieraan doen? Zo'n vraag vraagt om een antwoord, gezien het karakter van deze vraag zullen wij deze in onze 128 afdeling TRIPLE 128 publiceren! Op die manier heeft iedereen er wat aan. Er zijn immers nog genoeg mensen die met een gewone TV werken! Tot zover deze inleidende tekst, en zullen we dan nu maar snel beginnen met de hardware afdeling van TRIPLE 128? Lees verder en geniet van uw Commodore INFO. Wij zijn er immers niet voor niets!

Ons adres is:
Postbus 43048, 1009 ZA Amsterdam

P.S. Bij voorkeur dient u gebruik te maken van een getypte brief. Dit omdat niet iedereen het 'spijker-vrije' handschrift heeft.

In de hardware rubriek zullen wij voortaan onderdelen van de C128/D hardware toelichten. Tevens zullen er in de toekomst verschillende hardwareprojecten gaan verschijnen om de C128/D op eigen houtje uit te breiden en/of te verbeteren. Randapparatuur zoals diskdrive's, printers en monitoren zullen ook in deze rubriek worden getest. Met de resultaten die er in deze rubriek geplaatst zullen worden hopen wij u een beter zicht te geven op de vele verschillende fabrikaten en soorten randapparatuur. Wij zullen er tevens rekening mee houden dat er onder u personen zijn die het 'gevaarlijke' pad der computers nog maar net aan het bewandelen zijn. Tenslotte zijn we allemaal eens begonnen.

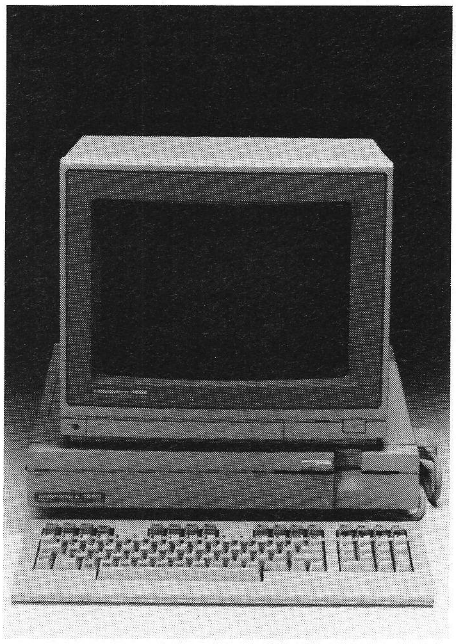
De C128/D

Het ontwerp van de Commodore 128 (afgekort C128) was geen gemakkelijke taak. De nieuwe computer moest ook de bestaande C64 programmatuur kunnen verwerken. Men heeft dit probleem opgelost door de computer in drie verschillende modi te laten draaien. En er moest ook CP/M software op de nieuwe computer kunnen draaien. Voor de CP/M mode werd de Z80B van Zilog op de tekentafel geplaatst. De C128 mode moest sneller en meer ram kunnen aanspreken. De 8502 microprocessor moest zowel de C64 als de C128 mode kunnen bedienden. Voor het gemak werd het weer een 8 bits computer.

Om goed met CP/M te kunnen werken werd er een 8563 VDC chip in het ontwerp opgenomen. Toen men toch bezig was moest het geen probleem meer wezen om het 80 cls ook in de gewone C128 mode te kunnen gebruiken. Om problemen te vermijden moest later de C64 80 cls mode verdwijnen. Men moest zich in de C64 mode maar weer met het 40 cls be-helpen. Voor de C64 mode koos men voor de oude basic/kernal ROM. Voor de C128 mode moest er een geheel nieuw en krachtig basic/kernal besturingssysteem worden ontwikkeld. Wie er voor de C64/C128/CP/M verantwoordelijk zijn moet u met sys32800,123,45,6 kunnen oproepen. Met uitzondering van de jaba /isan copy kernal. Wij maken er u op attent dat deze programmatuur door 'roof' is verkregen. U benadeelt bij aanschaf de programmeurs die deze software hebben ontwikkeld!

De processor

Er zijn twee microprocessors in de C128 aanwezig. Dit zijn de 8502 en de Z80B van Zilog. De eerste is geschikt voor zowel de C64 als de C128 mode. Deze processor draait op max. 2 Mhz. De microprocessors worden geklokt door de VICII 8564 chip. In de C64 mode op 1 mHz en in de CP/M mode 4 Mhz. In de C128 mode kunt u gebruik maken van zowel de 1 als de 2



Mhz mode. In basic (7.0) kunt u met het commando 'FAST' naar de 2 Mhz mode overschakelen. Met het basic commando 'SLOW' kunt u weer naar de 1 Mhz mode terugschakelen. Ook kunt u in plaats van fast **poke 53296,1** gebruiken. Voor slow kunt u **poke 53296,0** gebruiken. Voor het gemak vermelden wij hier dat het om adres $d030 gaat. Bij het gebruik van de 2 Mhz mode wordt het 40 cls uitgeschakeld. U kunt dan alleen gebruik maken van het 80 cls. Bij de 2 Mhz mode zal bij het aanspreken van de I/O componenten hardwarematig worden teruggeschakeld naar de 1 Mhz mode. Het 40 cls zal nog steeds niet beschikbaar zijn.

De modi

Het is met de C128 dus mogelijk om C64 software te draaien. Vaak alleen in de C64 mode en soms, met simpele basic programma's, in de C128 mode. Zie ook het artikel over conversie. Met het basic commando 'GO 64' kunt u de C64 mode oproepen. Ook de computer aanzetten en daarbij de Commodore toets ingedrukt houden werkt. Met sysdec ("e24b") kunt u de C128 in een C64 omtoveren. Er is nog een manier waarvan het programma Chessmaster 2000 gretig gebruik maakt. Dit programma is dus gewoon een C64 programma wat door middel van een programma in de boot sector overschakelt naar de C64. Zonder dat u het in de gaten heeft zit u in de C64 mode. Kijk maar eens met een diskmonitor op track 1, sector 1! En zo zijn er nog velen die op de bovenstaande manier de boel proberen te 'flessen'.

De behuizing

Als je de C128 voor het eerst ziet valt direct het enorme verschil met de C64 op. De computer is in zijn geheel in een zeer fraai uitgevoerde kast gestoken die tevens zeer gemakkelijk overal kan worden neergezet. Het toetsenbord is er vergeleken met dat van de C64, het zogenaamde haaientanden wasbord, zeer op vooruit gegaan, het slaat lekker licht aan en heeft een zeer lange levensduur. Ons toetsenbord 'leeft' na twee jaar nog steeds terwijl wij twee C64 toetsenborden per jaar nodig hadden. Zeer handig is het numerieke toetsenbord aan de rechterkant. Vooral als er zeer veel data ingetypt moet worden. Verder zijn de cursortoetsen zowel op de C64 plaats als een paar losse toetsen rechts boven het alfanummerieke toetsenbord uitgevoerd. De cursortoetsen en het extra keypad kunnen echter standaard niet in de C64 mode gebruikt worden maar hier valt met een aantal truuks wel wat aan te doen. In een volgend nummer zullen wij dit publiceren.

Tot 512 k extra RAM.

Er bestaat de mogelijkheid om via de cartridge poort het systeem met nog eens maximaal 512K RAM uit te breiden. Maximaal heeft u dus 640 KB en dat is net zo veel als een PC. Dit geheugen wordt door het systeem ingedeeld in blokken van 64K RAM. Deze 64K RAM wordt een bank genoemd, zodat we 'banking' toe moeten passen om het gehele geheugen te kunnen gebruiken (dit omdat het een 8 bits computer is en die kan maximaal 64

Kbyte adresseren). Gelukkig ondersteunt BASIC dit met behulp van het commando 'BANK'. Er zijn verschillende routines standaard in het geheugen aanwezig om data van het ene deel naar het andere deel van het geheugen te verplaatsen. De belangrijkste zijn de FETCH en STASH routines. Deze routines werken erg snel, en in de praktijk zal er bijna nooit op het verplaatsen van data gewacht hoeven te worden. Basic gebruikt bank 1 voor het opslaan van de variabelen. Zodat bank 0 gebruikt kan worden om een basic programma in op te slaan. Het is ook mogelijk om in bank 1 een tweede programma op te slaan. Op deze manier werken de programma's Superbase en Superscript samen. Het programma Superbase in bank 0 en Superscript in bank 1.

Zijn opvolger

De Commodore 128/D is een speciale uitvoering van de C128 die na de losse C128 op de markt kwam. De C128/D is in een systeemkast ondergebracht met een Commodore 1571 diskdrive. De print die zich in de C128/D bevindt is bijna identiek aan die van de 'losse' C128. De voeding is tevens in de kast ingebouwd zodat er meteen een snoer uit het stopcontact in de computer gestoken kan worden. Verder is het voordeel natuurlijk dat er geen lastige snoerenboel achter de computer ligt. Iets wat bij de losse versie van de C128 wel het geval is. Tevens beschikt de C128/D over een los toetsenbord dat met een kabel met de systeemkast verbonden is. Deze kabel is eigenlijk aan de korte kant. Er is een handvat aan de zijkant van de computer gemonteerd wat hem por-table (lees versleepbaar) maakt.

De aansluitingen

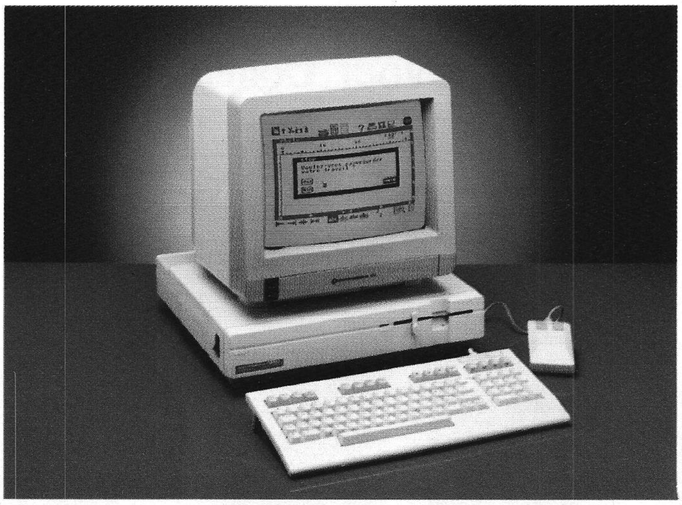
De aansluitingen voor de randapparatuur zijn voor de C128 en de C128/D natuurlijk hetzelfde (de print is identiek). Deze aansluitingen bestaan uit verschillende connectoren en zijn zo gekozen dat men de randapparatuur niet op verschillende (lees verkeerde) manieren kan aansluiten. We zullen de verschillende in- en uitgangen van de C128 in het kort bespreken.

De game-poorten

Deze poorten bevinden zich zowel bij de C128 als de C128/D aan de rechterkant van de computer. Hier bevindt zich ook de reset en aan/uit schakelaar. Bij de C128/D zit de aan/uit schakelaar aan de linkerkant. Ook zit hier bij de C128/D de drive reset knop verscholen. Deze knop is alleen met een lang en dun voorwerp te bedienen. Bij deze een tip! Wanneer de computer 'hangt' schakel dan niet direct de computer uit maar probeer eerst de reset knop. Dit omdat het vaak snel uit/aan schakelen van de C128/D schadelijke gevolgen kan hebben. De game poorten zijn bedoeld om joysticks, paddles of eventueel een muis op aan te sluiten. Al deze apparaten zijn voor de C128/D te verkrijgen. Tevens is het mogelijk om via deze game poorten twee A/D converters aan te spreken die zich in de geluidschip bevinden. Dit is echter zeker voor de beginner geen gemakkelijk stof. Wat je met joysticks kan doen weet iedereen wel.

De expansion poort

Deze poort die rechts achter op de computer zit is bedoeld om de computer later met hardware uit te breiden. Om zo bv een cartridge in te plaatsen (wie kent de Power Cartridge nu niet). Deze poort was ook al op de C64 aanwezig, en kan ook in de C128 mode gebruikt worden. De RAM-uitbreidingen voor een extra groot geheugen worden in deze poort gestoken. Zo is het mogelijk om bv extra ROM in deze poort te plaatsen. Viza Ltd. maakt gebruik van deze poort voor hun 'software beveiliging'.



De cassette poort

De cassettepoort was al op de PET van Commodore aanwezig. De C128/D had de eerste Commodore homecomputer kunnen worden die deze poort niet had. Deze cassettepoort is echter aangebracht om de compatibiliteit met de C64 te behouden. De meeste C128/D softwarepaketten ondersteunen de cassette recorder niet eens. Voor de C128 mode is deze poort totaal overbodig geworden. Het zou leuk zijn wanneer er iemand was die een programma had dat in de C128 mode werkt, en het mogelijk maakt om de cassetterecorder te gebruiken als 'tape streamer' voor backup doeleinden.

De seriële poort

Deze verschilt wel wezenlijk van de seriële bus die de C64 bezat, hij is in de C64 mode exact gelijk aan de seriële bus in een C64. Maar in de C128 wordt hij omgetoverd in een zogenaamde fast seriële bus wat zo'n 8 maal sneller werkt (in de CP/M mode zelfs zo'n 14 maal sneller). Hoe deze bus werkt zal in de nabije toekomst nog wel eens precies uitgelegd worden. De seriële bus is net als bij de C64 van het zogenaamde 'daisy chain' principe. Alle apparaten die aan de seriële bus worden aangesloten (printer en diskdrive) moeten worden 'doorgelust'. Dit houdt in dat er een kabel van de computer naar het apparaat loopt en een kabel van het apparaat naar het andere apparaat. Dit heeft als voordeel dat er niet zoveel draden uit de computer naar de randapparatuur hoeven te lopen. In één van de volgende nummers zullen wij een programma publiceren dat het mogelijk maakt om ook in de C64 mode met 1328 blocks op schijf te werken. Vooral de C64 B.B.S. freaks onder ons zullen dit waarderen. Met het door ons gepubliceerde soft/hardwareproject kunt u vanaf dan net zo snel in de C64 mode gebruik maken van de 1570/71 snelheid die gebruik maken van de snelle seriële bus.

Audio/video uitgang

Deze uitgang is geschikt voor de 40 cls display mode. U kunt dus zowel in de C64 als de C128 mode gebruik maken van deze uitgang. Omdat er echter een aparte videochip voor de 80 cls mode is gemaakt, is het mogelijk om in de C128 mode tegelijkertijd van zowel het 40 als het 80 cls gebruik te maken. Op deze uitgang staat tevens het

audiosignaal. Dit audiosignaal zorgt voor het geluid wat de SID chip kan produceren. Zo kunt u dit signaal aan de stereo installatie overdragen zodat het geluid veel beter wordt weergegeven. Deze uitgang noemen we ook wel de composite video uitgang.

HF uitgang

De HF uitgang is een hoogfrequent uitgang. Deze uitgang kan zonder problemen direct met een televisietoestel worden verbonden. Dit is het gemoduleerde 40 cls video en audio signaal.

RGBI uitgang

Deze uitgang is een enorme vooruitgang. De RGBI uitgang kwam niet op de C64 voor. Deze uitgang komt direct van de 80 cls videochip af en bevat de gescheiden Rood Groen en Blauw signalen. De benodigde synchronisatie signalen zijn natuurlijk ook op deze uitgang te vinden. De 80 cls videochip maakt de C128/D tot een professionele machine. Van deze 80 cls mode kan ook in de CP/M mode gebruik worden gemaakt. Met deze uitgang kunnen we over 16 kleuren beschikken. Dit omdat het hier geen analoge- maar digitale uitgang betreft. Over het hoe en wat komen wij later nog terug. U kunt dus een RGB monitor of een composite monitor op deze uitgang aansluiten. Voor het geluid zult u echter een verbinding moeten maken met de audio/video uitgang. In dit nummer zullen wij u uitleggen hoe u zelf zo'n schakelbare verbinding moet maken.

De userpoort

De userpoort is voor sommige mensen wel de meest belangrijke I/O verbinding die er op de computer zit. Op deze poort zit zowel een parallelle als een seriële poort. De seriële poort kan gebruikt worden om een seriële niet Commodore printer aan te sluiten. Ook kan er op deze seriële aansluiting een modem aangesloten worden. Deze seriële communicatie zullen wij in de nabije toekomst onder de loep nemen. Het parallelle gedeelte van de userpoort, kan gebruikt worden om printers aan te sturen die zijn uitgerust met een zogenaamd centronics interface. Deze printers kan men of via een programma of via een cartridge aansturen. Zo is het bijvoorbeeld met de Power Cartridge of Final Cartridge, en de benodige kabel, helemaal geen probleem meer. Ook kan men via deze userpoort robotjes besturen, dus voor de echte electronica fanaten kan de lol helemaal niet meer op. Over deze poort is dus bijna alles mogelijk. Er zijn snellaadsystemen op de markt die de laadtijd van de diskdrive met 35 maal versnellen. De overdracht gaat dan via de parallelle poort. Tot zover de aansluitingen van de C128/D.

De binnenkant

Natuurlijk is er binnen de computer ook heel wat veranderd. Zo is er nu meer RAM geheugen in de computer aanwezig. In de computer zitten ook meerdere besturings-systemen. Dit moet natuurlijk ook om in de C64 mode te kunnen werken. Er bevinden zich in de computer twee videochips. Dit zijn de 40 cls VICII chip en de 80 cls 8563 videochip. Voor de I/O zitten er twee CIA'S (6526) in de C128/D. Er een speciaal IC ontwikkeld om het geheugen te beheren. Onder de naam MMU (memory managment unit) is deze op de print aangebracht. De print ziet er erg verzorgd en degelijk uit. En dan komen we nu bij de Z80B processor van Zilog aan. Deze Z80 microprocessor heeft in de C128 de taak om met CP/M te kunnen werken.

CP/M op de C128

In de CP/M werkstand hebt u met de ingebouwde Z80 microprocessor alle mogelijkheden van de CP/M Plus versie 3.0 van Digital Research. Buiten de standaard commando's heeft Commodore er zelf ook nog wat aan toegevoegd. Met CP/M heeft u de beschikking over zowel het 40 als 80 cls. Maar het gebruik van het 40 cls is niet aan te raden. Met het CP/M Plus pakket van Commodore heeft de gebruiker de beschikking over 59 TPA's (Transient Program Areas). Het besturingssysteem wordt van de CP/M systeemdiskette ingelezen. U kunt dit doen door de CP/M systeem disk in de drive te plaatsen en de computer te resetten of uit/aan te schakelen. Ook met het basic commando 'BOOT' kunt u het CP/M Plus systeem inschakelen. De diskdrive's met type nummer 1570/1/2 zullen als het ware in de turbomode worden gezet. Met een RAM uitbreiding kunt u gebruik maken van een RAM disk. Met de door ons gebruikte RAM expansie module (1750/512K RAM) hebben wij de beschikking over een 512K RAM-DISK. Gezien de snelle interne verwerking bij het verplaatsen van data is dit supersonisch snel. Met zo'n RAM module heeft u nu als het ware een tweede diskdrive. Het nieuwe hiervoor benodigde CP/M systeem wordt standaard bij de module geleverd.

De penbezetting

Pen 1	Ground
Pen 2	Key
Pen 3	Restore
Pen 4	Plus 5 volt
Pen 5	Row 3
Pen 6	Row 6
Pen 7	Row 5
Pen 8	Row 4
Pen 9	Row 7
Pen 10	Row 2
Pen 11	Row 1
Pen 12	Row 0
Pen 13	Column 0
Pen 14	Column 6
Pen 15	Column 5
Pen 16	Column 4
Pen 17	Colomn 3
Pen 18	Colomn 2
Pen 19	Column 1
Pen 20	Column 7
Pen 21 t/m23	K0 t/m K2
Pen 24	40/80
25	Alpha Lock

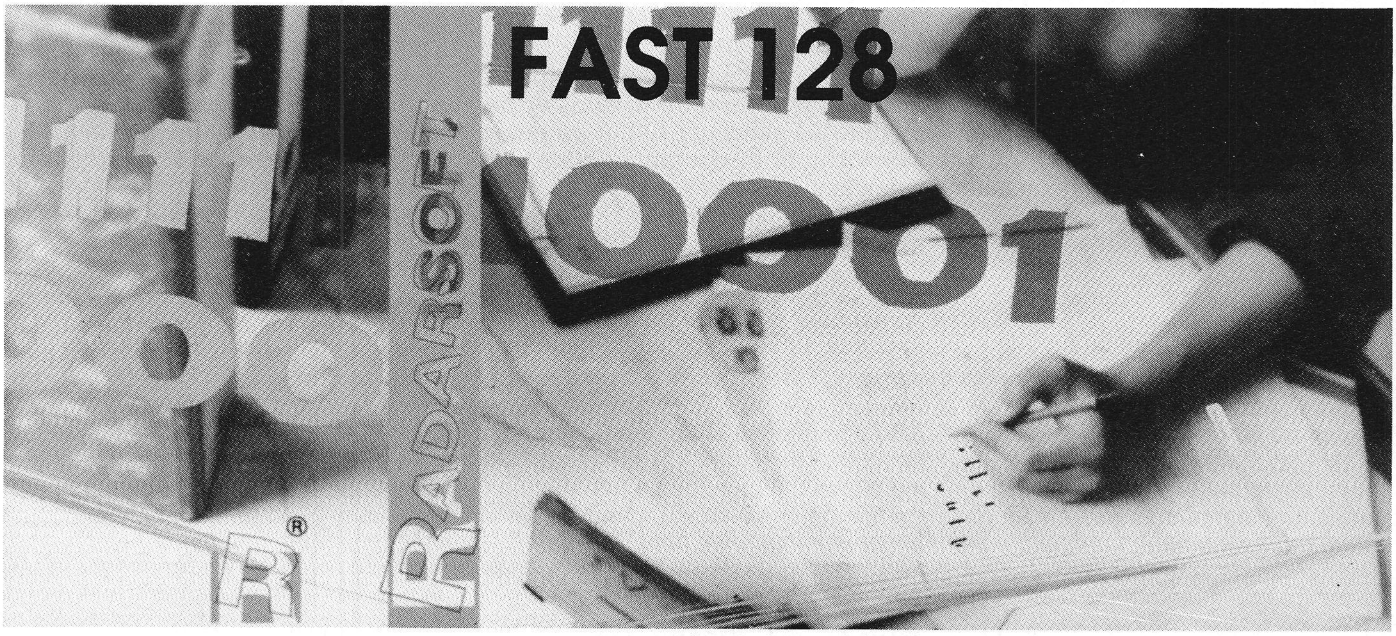
Marktprijs

Voor de personen onder ons die graag een C128 willen aanschaffen volgt hier een prijsindicatie. Een tweedehands computer verliest de eerste dag al 30% van de nieuwwaarde. Een losse C128 mag niet meer dan *f* 400,- kosten. De C128/D mag zo *f* 750,- kosten. Deze beide prijzen kunnen gebruikt worden als indicatie bij de aanschaf van een goed exemplaar zonder verdere randapparatuur.

Toetsenbord kabel

Het toetsenbord van de C128/D is met een kabel aan de computer verbonden. Omdat het niet zo'n geweldige lange kabel is volgt hier de oplossing. En wij vermelden er direct bij dat het echt handig is om zo'n lange kabel te bezitten! De bouw van deze kabelverlenger is door een ieder met alle gemak van de wereld te maken. Aan een soldeerbout en een stuk kabel heeft u bijna genoeg. Dat wil zeggen u dient nog twee sub D connectors aan te schaffen. Deze sub D's kosten zo'n *f*4.50 met kapje. Het is aan te raden om een kapje om de connector te monteren, die zijn los te koop, en kosten zo'n *f* 1,50 per stuk. Voor een kleine *f*17,50 heeft u een pracht van een kabel. Wij hebben u al verteld dat de bouw simpel is maar wat moet u doen. Gewoon, laten we maar draden zeggen, van de ene connector naar de andere aanleggen. Zo verbindt u pen één van de ene conector aan pen één van de andere conector. U dient in totaal 25 verbindingen te maken. De pen nummering is op de connector vermeld. Een kwestie van goed uit de doppen kijken dus want kortsluiting kost u een zogenaamde CIA (6526). Dat is dan *f* 50,00 ring ring kassa!

De software rubriek gebruiken wij elke aflevering voor het bespreken van een softwarepakket. De allereerste keer nemen wij een sofwarepakket van vaderlandse bodem onder de loep. Het pakket wordt door de niet geheel onbekende firma Radarsoft op de markt gebracht en heet FAST 128.



FAST 128 is een software ontwikkelingsprogramma dat in zowel de 40 als 80 kolommen mode werkt. FAST 128 werkt alleen in de C128 mode. Maar u kunt ook software voor andere (6502/6510/8502) georienteerde computers ontwikkelen. Zo kunnen we dus in de C128 mode software ontwikkelen voor bijvoorbeeld de C64. Het ontwikkelen zal veel sneller gaan omdat er voor de C64 niet zulke geweldige assemblers zijn. Met name de snelheid en opslag zijn hier in het geding. FAST 128 is een must voor alle machinetaal programmeurs. Fast is een two pass assembler/disassembler.

Voordelen

Stelt u zich eens voor, u wilt een stuk machinetaal ontwikkelen voor het een of ander. In de ontwikkelingsfase zet u, als u goed te werk gaat, alles op papier. Na deze ontwikkeling gaat u uw code invoeren met behulp van de ingebouwde monitor! Na het wegschrijven naar disk wilt u het programma even testen. Tijdens de test blijken er nog fouten in uw ontwerp te zitten! Nu komt het grote probleem van de monitor om de hoek, (daar zal menigeen het met ons over eens zijn). Het VERPLAATSEN/AANPASSEN van de zojuist ingevoerde machine-code. Alle routine's zullen moeten worden aangepast. En dat VREET tijd van heb ik jou daar. Dat dit bij grotere programma's helemaal geen doen meer is kan iedereen wel begrijpen. Dat wil zeggen, dit kost de gemiddelde hobby programmeur al gauw een paar dagen. Wij moeten er helemaal niet aan denken om in de ingevoerde machinetaal iets te moeten tussenvoegen. Stelt u zich eens voor, u heeft wat over het hoofd gezien en moet 5 rottige byte's toevoegen! U kunt mischien uw routine's nog wat inkorten, en zo over uw programma heen JUMPEN naar een andere subroutine, maar dat is water naar de zee dragen! En vaak is het helemaal niet eens mogelijk. In FAST 128 had u nu gewoon een regel (of meerdere) tussengevoegd. En daarmee was de kous af. Even door de assembler heen, wegschrijven en weer testen. Programmeur Lex Boere kan daar vast over meespreken! En gezien het aantal goede assemblers (0,0) naast FAST 128 heeft hij (waarschijnlijk) gedacht 'daar maak ik een eind aan!'

FAST en Triple 128

In het vervolg wanneer wij een stuk machinetaal plaatsen, zullen wij er voor zorgen dat er ook een 'FAST listing' in het blad wordt afgedrukt! Alweer een reden om FAST 128 in de software-bak te halen. Hierbij vragen wij de personen die een machinetaal routine naar ons willen opsturen a.u.b. een FAST 128 listing bij te voegen. Daar kunt u dan ook tekst bij plaatsen waardoor we de routine niet eerst moeten bestuderen.

En de monitor dan?

Alle andere assemblers en monitoren kunnen na het verschijnen van FAST 128 de prullebak in. Waarom? Hier volgt een tweede voorbeeld. U heeft al geruime tijd terug een routine ontwikkeld. U moet die nu aanpassen. Dat is in de monitor, na langere tijd, niet zo eenvoudig. U zult op de bovenkamer moeten vertrouwen. Maar wat als de grijze cellen het laten afweten? Met FAST 128 heeft u daar geen last van, gewoon uw programma van tekst en uitleg voorzien! Op deze manier weet u tenminste na enige tijd ook nog wat u met een routine bedoelde. Wij spreken uit ervaring, en u kunt van ons aannemen dat zo'n assembler fantastisch is. Wij hebben zelf wel eens het volgende gezegd. Citaat: 'Wat doen 'ZE' hier nu weer' Die 'ZE' waren wij zelf. Na een jaar waren we de gehele routine al weer vergeten, en onze documentatie was verloren gegaan. Een grote frustratie als je met zogenaamde 'DEADLINE'S' te maken hebt. Door het aanschaffen van FAST 128 kunt U die frustratie voorkomen. Neem van ons aan dat je dat niet lekker zit, wanneer een collega programmeur je er op moet attenderen wat je zojuist hebt gezegd. Wij hadden bij hem het idee gewekt dat de routine was 'geleend'. U voelt hem al aankomen dachten wij zo. Wij weten het zo'n monitor hoef je niet in te laden maar is bij lange na niet zo krachtig als FAST 128. Bovendien, dat inladen is ook geen probleem, en gaat bovendien snel genoeg.

De FAST editor

U bent als het ware in een editor aan het werk. En dit zal voor de aspirant machinetaal programmeur een hele opluchting zijn. Er zijn drie invoermodi: de gewone basic mode, de mnemonics mode om assembly instructies kompakt op te kunnen slaan en als laatste de tekst mode. Fast is dus een soort editor zoals de basic interpreter maar dan in een hele luxe vorm. Het uiterlijk is niet fraai maar u kunt nu machine-code invoeren. Fast maakt gebruik van regels om de code op te plaatsen. Na het invoeren kunt u de 'broncode' zoals deze wordt genoemd, vertalen met de assembler. Ook kunt u met Fast bestaande machine-code vertalen in zo'n broncode.

Basic en FAST

U kunt natuurlijk ook gewoon in basic programmeren en waar nodig in machinetaal. Waar nodig? Ja, waar basic te traag is, en het dus laat afweten, grijpt u met machinetaal in! Ook weer een voordeelvan FAST 128 dat als één van de weinige met basic kan samenwerken. Dus voor de persoon die in machinetaal wil gaan werken is FAST 128 de stap in de goede richting.

Het gebruik

Natuurlijk zullen er al vele personen met deze asssembler aan het werk zijn en is het inmiddels gesneden koek. Maar voor de personen die nog niet aan de machinetaal zijn toegekomen is nu het moment aangebroken voor dit prachtige stuk gereedschap. Met het commando 'bnk' kunnen we er voor zorgen dat de code in die betrefende bank zal draaien. De regels waarop de code staat kunnen we eenvoudig hernummeren. We kunnen met FAST 128 gebruik maken van 'labels' en door die 'labels' kunnen we erg snel te werk gaan. Ook kunnen we subroutine's heel eenvoudig aanroepen. Wat dacht u van 'jsr invoer', om een karakter op te halen, dat is heel eenvoudig vergeleken bij 'jsr$2345'. Want laten we nu eerlijk zijn 'jsr invoer' is en blijft beter te onthouden. Op deze manier kunnen we een hele bibliotheek opbouwen met subroutine's die u later weer kunt gebruiken bij het ontwikkelen van andere software. Dus die labels zijn een hele kracht bij het onwikkelen van uw software. U kunt een sourcefile achter de bestaande file in het geheugen aan elkaar plakken. Het opzoeken van 'labels' en opcode's is nu ook gemakkelijk te doen. Er een commando waarmee u kunt zoeken en vervangen. De tab positie's kunnen ook gemakkelijk worden ingesteld. In totaal zijn er 28 verschrikkelijk verslavende commando's in FAST 128 te gebruiken. Fast 128 is een super snel programma dat in 'no-time' de benodigde code voor u in het geheugen plaatst. En over opslag heeft u ook niet te klagen! Zo'n 52K byte's kunnen als broncode worden ingevoerd. Dat zijn wel 209 blokken op uw diskette! En heeft u wel eens zoveel machinetaal achter elkaar gebruikt? Vast niet, maar daarvoor is FAST 128 wel geschikt. Waarschijnlijk ook omdat de Radarsoft programmeurs zelf ook zo'n assembler nodig hadden. En onder het motto als je toch bezig bent Lex!

Nadeel

Een groot nadeel is volgens ons dat je niet vanuit FAST 128 zelf de zojuist geassembleerde code kan wegschrijven, zonder dat je gebruik moet maken van bsave bla,bla of direct de monitor moet induiken. Dit zou volgens ons een oplossing kunnen wezen, na het assembleren de pointers van het start en eindadres gebruiken bij het wegschrijven van de machine-code!

```
100 ; VOORBEELD PROGRAMMA FAST 128
110 ;
120                 BNK 0              ; Data in bank 1
130                 ORG $0B00          ; Data op adres $0b00
140 ;
150 NORMIRQ         EQU $FA65          ;
160 BORDER          EQU $D020          ; Declaraties
170 LOVEC           EQU $0314          ;   labels
180 HIVEC           EQU LOVEC+1        ;
190 ;
200                 LDA #<NIEUWEIRQ    ;
210                 STA LOVEC          ; Vector naar eigen
220                 LDA #>NIEUWEIRQ    ;  irq routine opzetten
230                 STA HIVEC          ;
240                 RTS
250 ;
260 NIEUWEIRQ       INC BORDER         ; Verhoog borderkleur
270                 JMP NORMIRQ        ; Vervolg normale irq
280 ;
290                 END
```

De praktijk

Kort voor het schrijven van deze tekst waren wij in Hannover op de 'CeBIT 88' en wat we daar meemaakten was werkelijk de klap op de vuurpijl. Wij kwamen daar twee personen van de MASTER GENIUS CRACKING CREW tegen, die verklaarden daar in minder dan een half uur een demo te kunnen programmeren, die de daar getoonde Commodore Amiga 500 met zijn demo in 'HIGH-RES' in verlegenheid zou brengen. Natuurlijk was er niemand die deze uitspraak wou geloven. De heren voelden zich op hun tenen getrapt, en moesten zich bewijzen. Na een uur was er een C128/D ,met een dozijn leds en schakelaars, op tafel gezet. Een heel ander opstartscherm kwam te voorschijn. Men schakelde over op, het was bijna niet te geloven, 'FAST 128'. Zij waren er blijkbaar in geslaagd om 'FAST 128' op 'ROM' te zetten. Men had 32 Kbyte ROM extra in de C128! Na veel praten werd een op de beurs geleende Nec Multisync aan de C128/D gekoppeld. En na zo'n 10 minuten werd er al getest. Na 13 minuten was het besturingssysteem aangepast. Na 15 minuten kon de picture worden overgezet! Na 19 minuten stond de picture die op de Amiga stond, met een veel hogere resolutie op het C128 scherm! Hiermee verdiende deze heren 'even' hun brandstof en het nodige voedsel terug! Alle Duitsers werden door deze beide heren sprakeloos. Er werd daar zelfs gezegd, "En daar heb ik een Amiga voor moeten kopen, nu blijkt dat 'ding' het ook te kunnen". Bij navraag bleek dat zonder hun versie van FAST 128, het NOOIT en te nimmer in die korte tijd zou zijn gelukt. Maar het zweet stond hun dan ook op het voorhoofd. De heer van Markt & Technik moest onder veel Duits gejubel zijn zojuist verloren weddenschap uitbetalen. Het bovenstaande verhaal ondersteunt de ware kracht van het pakket.

Onze conclusie!

Een geweldig, supersnel, handig en buitengewoon gebruikersvriendelijk programma met professionele trekjes! Een absolute must voor de machinetaal programmeur, of zij die er over denken om in deze taal te gaan werken. Voor de prijs hoeft u het niet te laten. Dit programma heeft bij ons na er mee te hebben gewerkt, een zee van rust en perfectie opgeroepen! Bij deze willen wij de firma Radarsoft bedanken voor het ter beschikking stellen van dit 'SUPER' pakket.

RicoGraphic C 128

Omdat dit programma wat Rico van Eeten ons opstuurde nogal wat mogelijkheden heeft, is er een menu opgenomen. Druk op het pijltje links boven om daarin te komen vervolgens op 1 en u krijgt het scherm met alle mogelijkheden. De grafische commando's hebben standaardwaarden die uitgaan van de positie van de pijl en werken daardoor erg gemakkelijk. Nodig zijn: een joystick, een diskdrive 15xx en natuurlijk een Commodore 128 (D)

```
94    rem        ***  ricographic  ***
95    :
96    rem  een grafisch progamma voor c1
      28
97    :
98    rem    gemaakt door rico van eeten
99    :
100   if peek(215)<>0then print"dit[SPA
      CE]is[SPACE]een[SPACE]40[SPACE]kar
      akters[SPACE]c128-progamma[SPACE]!
      ":end
101   k0=16:k1=1:k4=13:ks=13:gosub111:gr
      aphic1,1:graphic0,1
102   trap 600
103   fork=3584to3614:pokek,o:nextk
104   fork=3615to3710:reada:pokek,a:next
      k
105   goto300
109   :
110   sprite1,1,ks:sprite2,1,ks:return
111   color0,k0:color1,k1:color4,k4:grap
      hic1:return
112   fork=1to39:print"-";:nextk
113   window0,2,28,20
114   print"kleur:[SPACE]0=achtergr.[SPA
      CE]1=voorgr."
115   return
116   window29,2*w-1,39,2*w+1:printchr$(
      17)"[SPACE]"dchr$(145)chr$(145):in
      puta(w):return
199   :
200   color0,1
201   color4,1
202   color5,14
203   graphic0,1
204   char1,11,1,"[2xSPACE]***[SPACE]men
      u[SPACE]***[2xSPACE]",1
205   char1,6,6,"[SPACE]1[2xSPACE]menu[S
      PACE]hires[SPACE]beeldscherm[SPACE
      ]",1
206   char1,6,10,"[SPACE]2[2xSPACE]hires
      [SPACE]beeldscherm[6xSPACE]",1
207   char1,6,14,"[SPACE]3[2xSPACE]grafi
      sche[SPACE]comando's[4xSPACE]",1
208   char1,6,18,"[SPACE]4[2xSPACE]disk-
      comando's[9xSPACE]",1
209   fork=1to10:geta$:nextk:getkeya$
210   ifa$="1"then215
211   ifa$="2"thengosub110:gosub111:goto
      304
212   ifa$="3"then400
213   ifa$="4"then500
214   goto209
215   scnclr
216   printspc(4)chr$(18)"[SPACE]***[SPA
      CE]menu[SPACE]hires[SPACE]beeldsch
      erm[SPACE]***[SPACE]"chr$(146)
217   print:print"joystic[SPACE]in[SPACE
      ]poort[SPACE]2"
218   print"pijltjestoetsen[SPACE]voor[S
      PACE]snelheid"
219   print:printchr$(18)"*a"chr$(146)"[
      SPACE]achtergrondkleur[2xSPACE]"ch
      r$(18)"*v"chr$(146)"[SPACE]voorgro
      ndkleur"
220   print:printchr$(18)"*b"chr$(146)"[
      SPACE]borderkleur[7xSPACE]"chr$(18
      )"*p"chr$(146)"[SPACE]pijlkleur"
221   print:printchr$(18)"*1-8"chr$(146)
      "[SPACE]puntkleuren[5xSPACE]"chr$(
      18)"c=[SPACE]1-8"chr$(146)"[SPACE]
      puntkleuren"
222   print:printchr$(18)"*c"chr$(146)"[
      SPACE]puntcursor[SPACE]breed[2xSPA
      CE]"chr$(18)"*d"chr$(146)"[SPACE]p
      untcursor[SPACE]smal"
223   print:printchr$(18)"home"chr$(146)
      "[SPACE]puntcursor[SPACE]linksbove
      n"
224   print:printchr$(18)"clr"chr$(146)"
      [SPACE]scherm[SPACE]schoon[4xSPACE
      ]"chr$(18)"*o"chr$(146)"[SPACE]kle
      uropvulling"
225   print:printchr$(18)"del"chr$(146)"
      [SPACE]puntje[SPACE]in[SPACE]achte
      rgrondkleur[SPACE]zetten"
226   print:printchr$(18)"+"chr$(146)"[S
      PACE]puntcursor[SPACE]in[SPACE]voo
      rgrondkleur"
227   print:printchr$(18)"-"chr$(146)"[S
      PACE]puntcursor[SPACE]in[SPACE]ach
      tergrondkleur"
228   print:printchr$(18)" "chr$(146)"[S
      PACE]naar[SPACE]hoofdmenu"
229   char1,0,24,"[SPACE]'*'=control-toe
      ts[5xSPACE]druk[SPACE]een[SPACE]to
      ets[SPACE]",1
230   fork=1to10:geta$:nextk:getkeya$
231   gosub110:gosub111:goto303
299   :
300   gosub110:gosub111
301   locate 160,100
302   cs=1
303   x=rdot(0):y=rdot(1)
304   movespr1,x+13,y+41
305   j2=joy(2)
306   ifj2>127thendrawcs,x,y:j2=j2-128
307   ifj2>0thenonj2goto320,322,324,326,
      328,330,332,334
308   geta$:ifa$=""then305
309   a=asc(a$):ifa<20then337
310   ifa<32then347
311   ifa=45thencs=0:goto303
312   ifa=43thencs=1:goto303
313   ifa=95thensprite1,0:sprite2,0:goto
      200
314   ifa=129thencolor1,9:k1=9:goto303
315   ifa=144thencolor1,1:k1=1:goto303
316   ifa=145thenify>8thenlocatex,y-8:go
      to303
317   ifa>146thenifa<160thenona-146goto3
      54,355,356,357,358,359,360,361,362
      ,363,364,366,367
318   goto303
319   :
320   ify>0thenlocatex,y-1
321   goto303
322   ify>0thenifx<319thenlocate x+1,y-1
323   goto303
```

```
324  ifx<319thenlocatex+1,y
325  goto303
326  ifx<319thenify<199thenlocatex+1,y+
     1
327  goto303
328  ify<199thenlocatex,y+1
329  goto303
330  ifx>0thenify<199thenlocatex-1,y+1
331  goto303
332  ifx>0thenlocatex-1,y
333  goto303
334  ifx>0thenify>0thenlocatex-1,y-1
335  goto303
336  :
337  ifa=1then368
338  ifa=2thenk4=int(rnd(0)*16+1)color4
     ,k4:goto303
339  ifa=3thenwith2:goto303
340  ifa=4thenwith1:goto303
341  ifa=5thencolor1,2:k1=2:goto303
342  ifa=15thenpaintcs,x,y:goto303
343  ifa=16thenks=int(rnd(1)*15+1):gosu
     b110:goto303
344  ifa=17thenify<191thenlocatex,y+8:g
     oto303
345  ifa=19thenlocate0,0:goto303
346  goto303
347  ifa=20thendraw0,x,y:goto303
348  ifa=22then373
349  ifa=28thencolor1,3:k1=3:goto303
350  ifa=29thenifx<312thenlocatex+8,y:g
     oto303
351  ifa=30thencolor1,6:k1=6
352  ifa=31thencolor1,7:k1=7
353  goto303
354  locate0,0:scnclr1
355  goto303
356  color1,10:k1=10:goto303
357  color1,11:k1=11:goto303
358  color1,12:k1=12:goto303
359  color1,13:k1=13:goto303
360  color1,14:k1=14:goto303
361  color1,15:k1=15:goto303
362  color1,16:k1=16:goto303
363  color1,5:k1=5:goto303
364  ifx>8thenlocatex-8,y
365  goto303
366  color1,8:k1=8:goto303
367  color1,4:k1=4:goto303
368  movespr2,99,99:rd=int(rnd(1)*16):k
     0=rd:color0,k0+1:k=7168
369  geta$:ifa$=chr$(1)then368
370  pokek,rd+(peek(k)and240)
371  ifk<>8167thenk=k+1:goto369
372  movespr2,0,0:goto303
373  movespr2,99,99:rd=int(rnd(1)*16)*1
     6:k1=rd/16+1:color1,k1:k=7168
374  geta$:ifa$=chr$(22)then373
375  pokek,rd+(peek(k)and15)
376  ifk<>8167thenk=k+1:goto374
377  movespr2,0,0:goto303
399  :
400  scnclr:printchr$(18)"[SPACE]***[SP
     ACE]grafische[SPACE]comando's[SPAC
     E]***[SPACE]"chr$(146)
401  print:print:print"1[SPACE]box[5xSP
     ACE]vierkanten[SPACE]tekenen"
402  print:print:print"2[SPACE]char[4xS
     PACE]karakters[SPACE]op[SPACE]'t[S
     PACE]scherm[SPACE]zetten"
403  print:print:print"3[SPACE]circle[2
     xSPACE]cirkels[SPACE]en[SPACE]ande
     re[SPACE]figuren"
404  print:print:print"4[SPACE]draw[4xS
     PACE]punten[SPACE]en[SPACE]lijnen[
     SPACE]tekenen"
405  fork=1to10:geta$:nextk:getkeya$
406  a=val(a$):ifa>0anda<5thenonagoto40
     8,426,439,459
407  goto200
408  scnclr:printspc(18)"box"
409  gosub112
410  print:print"linkerbovenhoek[SPACE]
     x[SPACE](0-319)"
411  print:print"[16xSPACE]y[SPACE](0-1
     99)"
412  print:print"rechteronderhoek[SPACE
     ]x[SPACE](0-319)"
413  print:print"[17xSPACE]y[SPACE](0-1
     99)"
414  print:print"rotatie[SPACE](0-360)"
415  print:print"inkleuren?[SPACE]0=nee
     [SPACE]1=ja"
416  w=1:d=1:gosub116
417  ifx-7>0thenx1=x-7:elsex1=0
418  ify-7>0theny1=y-7:elsey1=0
419  w=2:d=x1:gosub116
420  w=3:d=y1:gosub116
421  w=4:d=x:gosub116
422  w=5:d=y:gosub116
423  w=6:d=0:gosub116
424  w=7:d=0:gosub116
425  goto470
426  scnclr:printspc(18)"char"
427  gosub112
428  print:print"karakterkolom[SPACE](0
     -39)"
429  print:print"karakterrij[SPACE](0-2
     4)"
430  print:print"omgekeerde[SPACE]kl.?[
     SPACE]0=nee[SPACE]1=ja"
431  w=1:d=1:gosub116
432  w=2:d=int(x/8):gosub116
433  w=3:d=int(y/8):gosub116
434  w=4:d=0:gosub116
435  window0,15,39,24
436  print"typ[SPACE]achter[SPACE]'?'[S
     PACE]de[SPACE]af[SPACE]te[SPACE]dr
     ukken[SPACE]tekst[3xSPACE]geen[SPA
     CE]comma's[SPACE]e.d.[SPACE]gebrui
     ken!"
437  inputt$
438  goto470
439  scnclr:printspc(15)"circle"
440  gosub112
441  print:print"middelpunt[SPACE]x[SPA
     CE](0-319)"
442  print:print"[11xSPACE]y[SPACE](0-1
     99)"
443  print:print"radius[SPACE]x"
444  print:print"[7xSPACE]y"
445  print:print"starthoek[SPACE](0-360
     )"
446  print:print"eidhoek"
447  print:print"rotatie[SPACE](0-360)"
448  print:print"graden[SPACE]tussen[SP
     ACE]segmenten"
449  w=1:d=1:gosub116
450  w=2:d=x:gosub116
451  w=3:d=y:gosub116
452  w=4:d=30:gosub116
453  w=5:d=a(4):gosub116
```

```
454  w=6:d=0:gosub116
455  w=7:d=360:gosub116
456  w=8:d=0:gosub116
457  w=9:d=1:gosub116
458  goto470
459  scnclr:printspc(18)"draw"
460  gosub112
461  print:print"punt[SPACE]x[SPACE](0-
     319)"
462  print:print"[5xSPACE]y[SPACE](0-19
     9)"
463  print:print"naar[SPACE]punt[SPACE]
     x[SPACE](0-319)"
464  print:print"[10xSPACE]y[SPACE](0-1
     99)"
465  w=1:d=1:gosub116
466  w=2:d=x:gosub116
467  w=3:d=y:gosub116
468  w=4:d=a(2):gosub116
469  w=5:d=a(3):gosub116
470  printchr$(19)+chr$(19)
471  char1,27,23,"okee[SPACE](j/n)?",1
472  fork=1to10:getb$:next:getkeyb$
473  ifb$<>"j"then200
474  gosub110
475  gosub111
476  ifa$="1"thenboxa(1),a(2),a(3),a(4)
     ,a(5),a(6),a(7)
477  ifa$="2"thenchara(1),a(2),a(3),t$,
     a(4)
478  ifa$="3"thencirclea(1),a(2),a(3),a
     (4),a(5),a(6),a(7),a(8),a(9)
479  ifa$="4"thendrawa(1),a(2),a(3) to
     a(4),a(5)
480  goto303
499  :
500  scnclr:printspc(8)chr$(18)"[SPACE]
     ***[SPACE]disk-comando's[SPACE]***
     [SPACE]"chr$(146)
501  print:print"1[SPACE]directory"
502  print"2[SPACE]bitmap[SPACE]laden[S
     PACE]van[SPACE]disk"
503  print"3[SPACE]bitmap[SPACE]saven[S
     PACE]naar[SPACE]disk"
504  fork=1to10:geta$:nextk:getkeya$
505  ifa$="1"then508
506  ifa$="2"ora$="3"then511
507  goto200
508  window6,5,39,21,1
509  directory:printchr$(19)chr$(19)
510  goto504
511  window4,21,36,23,1
512  print"typ[SPACE]achter[SPACE]'?'[S
     PACE]de[SPACE]filename"
513  inputb$
514  iflen(b$)>15then511
515  printchr$(19)chr$(19):scnclr
516  ifa$="2"thenprint"bload"+chr$(34)+
     b$+chr$(34)
517  ifa$="3"thenprint"bsave"+chr$(34)+
     b$+chr$(34)+",p7168[SPACE]to[SPACE
     ]p16192"
518  printchr$(17)chr$(17)chr$(17)chr$(
     17)"run523"chr$(145)chr$(145)chr$(
     145)chr$(145)chr$(145)
519  fork=842to844:pokek,145:nextk
520  fork=845to847:pokek,13:nextk
521  poke208,6
522  end
523  k1=(peek(7667)and240)/16+1
524  k0=(peek(7667)and15)+1
525  k4=13:ks=13:goto300
599  :
600  ifel=103orel=104thenscnclr:print"f
     out[SPACE]in[SPACE]dato[SPACE]rege
     ls[SPACE]103,104,700-711"
601  ifel>208andel<215thenresume209
602  ifel=230thenresume230
603  ifel>299andel<400thenresume300
604  ifel>404andel<408thenresume404
605  ifel>407andel<426thenresume408
606  ifel>425andel<439thenresume426
607  ifel>438andel<459thenresume439
608  ifel>458andel<470thenresume459
609  ifel>469andel<500thenresume303
610  ifel>499andel<600thenresume500
611  printchr$(19)chr$(19)err$(er)"[SPA
     CE]in[SPACE]line[SPACE]"el
612  end
699  :
700  data 15,128,,14,,,15,,,9,128,,8
701  data 192,,,96,,,48,,,24,,,12
702  data 0,,6,,,3,,255,255,255,128,,1
703  data 142,61,225,179,97,153,176,97
704  data 153,183,121,159,179,97,153
705  data 179,97,153,140,61,225,128,,1,
     128,,1,179
706  data 97,225,179,97,153,179,97,153
707  data 179,97,153,179,97,153,179,97
708  data 153,158,125,225,128,,1,255
709  data 255,255,255,255,255
710  data 153,158,125,225,128,,1,255
711  data 255,255,255,255,255
```

** EINDE LISTING ricgraph

REGEL 94	182	REGEL 200	116	REGEL 221	116	REGEL 309	206	REGEL 330	198
REGEL 95	58	REGEL 201	120	REGEL 222	152	REGEL 310	41	REGEL 331	31
REGEL 96	38	REGEL 202	173	REGEL 223	113	REGEL 311	95	REGEL 332	10
REGEL 97	58	REGEL 203	107	REGEL 224	13	REGEL 312	94	REGEL 333	31
REGEL 98	64	REGEL 204	250	REGEL 225	248	REGEL 313	71	REGEL 334	82
REGEL 99	58	REGEL 205	161	REGEL 226	82	REGEL 314	56	REGEL 335	31
REGEL 100	233	REGEL 206	152	REGEL 227	197	REGEL 315	37	REGEL 336	58
REGEL 101	143	REGEL 207	11	REGEL 228	78	REGEL 316	50	REGEL 337	247
REGEL 102	109	REGEL 208	220	REGEL 229	251	REGEL 317	42	REGEL 338	215
REGEL 103	98	REGEL 209	52	REGEL 230	52	REGEL 318	31	REGEL 339	253
REGEL 104	78	REGEL 210	86	REGEL 231	210	REGEL 319	58	REGEL 340	253
REGEL 105	28	REGEL 211	146	REGEL 299	58	REGEL 320	11	REGEL 341	195
REGEL 109	58	REGEL 212	84	REGEL 300	121	REGEL 321	31	REGEL 342	98
REGEL 110	189	REGEL 213	86	REGEL 301	58	REGEL 322	192	REGEL 343	190
REGEL 111	201	REGEL 214	36	REGEL 302	121	REGEL 323	31	REGEL 344	100
REGEL 112	69	REGEL 215	232	REGEL 303	242	REGEL 324	120	REGEL 345	90
REGEL 113	202	REGEL 216	204	REGEL 304	91	REGEL 325	31	REGEL 346	31
REGEL 114	0	REGEL 217	50	REGEL 305	128	REGEL 326	52	REGEL 347	254
REGEL 115	142	REGEL 218	246	REGEL 306	167	REGEL 327	31	REGEL 348	38
REGEL 116	67	REGEL 219	198	REGEL 307	30	REGEL 328	127	REGEL 349	250
REGEL 199	58	REGEL 220	189	REGEL 308	101	REGEL 329	31	REGEL 350	97

REGEL 351	160	REGEL 405	52	REGEL 438	36	REGEL 471	127	REGEL 522	128
REGEL 352	163	REGEL 406	45	REGEL 439	230	REGEL 472	235	REGEL 523	160
REGEL 353	31	REGEL 407	27	REGEL 440	33	REGEL 473	28	REGEL 524	91
REGEL 354	197	REGEL 408	32	REGEL 441	73	REGEL 474	31	REGEL 525	217
REGEL 355	31	REGEL 409	33	REGEL 442	90	REGEL 475	32	REGEL 599	58
REGEL 356	199	REGEL 410	185	REGEL 443	208	REGEL 476	17	REGEL 600	45
REGEL 357	201	REGEL 411	90	REGEL 444	9	REGEL 477	217	REGEL 601	10
REGEL 358	203	REGEL 412	255	REGEL 445	172	REGEL 478	1	REGEL 602	117
REGEL 359	205	REGEL 413	90	REGEL 446	169	REGEL 479	167	REGEL 603	8
REGEL 360	207	REGEL 414	15	REGEL 447	15	REGEL 480	31	REGEL 604	9
REGEL 361	209	REGEL 415	218	REGEL 448	233	REGEL 499	58	REGEL 605	16
REGEL 362	211	REGEL 416	250	REGEL 449	250	REGEL 500	9	REGEL 606	20
REGEL 363	111	REGEL 417	60	REGEL 450	34	REGEL 501	150	REGEL 607	30
REGEL 364	25	REGEL 418	64	REGEL 451	36	REGEL 502	64	REGEL 608	27
REGEL 365	31	REGEL 419	83	REGEL 452	47	REGEL 503	151	REGEL 609	11
REGEL 366	117	REGEL 420	85	REGEL 453	147	REGEL 504	52	REGEL 610	14
REGEL 367	109	REGEL 421	36	REGEL 454	254	REGEL 505	91	REGEL 611	236
REGEL 368	39	REGEL 422	38	REGEL 455	104	REGEL 506	148	REGEL 612	128
REGEL 369	115	REGEL 423	254	REGEL 456	0	REGEL 507	27	REGEL 699	58
REGEL 370	66	REGEL 424	255	REGEL 457	2	REGEL 508	51	REGEL 700	107
REGEL 371	63	REGEL 425	36	REGEL 458	36	REGEL 509	197	REGEL 701	211
REGEL 372	71	REGEL 426	85	REGEL 459	101	REGEL 510	34	REGEL 702	204
REGEL 373	81	REGEL 427	33	REGEL 460	33	REGEL 511	94	REGEL 703	6
REGEL 374	162	REGEL 428	161	REGEL 461	154	REGEL 512	232	REGEL 704	157
REGEL 375	18	REGEL 429	254	REGEL 462	90	REGEL 513	235	REGEL 705	11
REGEL 376	59	REGEL 430	202	REGEL 463	188	REGEL 514	90	REGEL 706	20
REGEL 377	71	REGEL 431	250	REGEL 464	90	REGEL 515	191	REGEL 707	28
REGEL 399	58	REGEL 432	13	REGEL 465	250	REGEL 516	96	REGEL 708	135
REGEL 400	54	REGEL 433	15	REGEL 466	34	REGEL 517	166	REGEL 709	63
REGEL 401	158	REGEL 434	252	REGEL 467	36	REGEL 518	230	REGEL 710	135
REGEL 402	49	REGEL 435	4	REGEL 468	144	REGEL 519	73	REGEL 711	63
REGEL 403	199	REGEL 436	14	REGEL 469	146	REGEL 520	25		
REGEL 404	28	REGEL 437	253	REGEL 470	71	REGEL 521	147		

Catch-it 128 .

De bedoeling van dit spel van W. Versloot is de bal op te vangen met een auto. Om het geheel wat moeilijker te maken kan de auto niet stoppen.

```
10   b1=bump(1):b2=bump(2):restore
20   fast:gosub250
30   gosub110:gosub340
40   movespr2,131+int(rnd(0)*97)+1#l-4
50   j=joy(2):ifj=3thenmovespr1,90#l
60   ifj=7thenmovespr1,270#l
70   ifj=128thenmovespr1,0#0
80   ifbump(2)=2thengosub130
90   ifbump(1)=3thengosub160
100  goto50
110  char0,4,12,"[COM-6]press[SPACE]fir
     e[SPACE]button[SPACE]for[SPACE]nex
     t[SPACE]level":do:loopuntiljoy(2)=
     128
120  char0,4,12,"[32xSPACE]":return
130  movespr2,0#0:sprite2,0:movespr2,17
     1,83:sound1,500,10:t=t-1
140  t$=str$(t):char0,36,2,t$:ift=0then
     goto200
150  sprite2,1:movespr2,131+int(rnd(0)*
     97)+1#l-4:b2=bump(2):return
160  movespr2,0#0:sprite2,0:s=s+1:moves
     pr2,171,83:sound1,10000,8
170  s$=str$(s):char0,8,2,s$
180  r=r+1:ifr=20thenr=0:l=l+1:l$=str$(
     l-4):char0,23,2,l$:gosub110
190  sprite2,1:movespr2,131+int(rnd(0)*
     97)+1#l-4:b1=bump(1):return
200  char0,10,9,"[SPACE]g[SPACE]a[SPACE
     ]m[SPACE]e[2xSPACE]o[SPACE]v[SPACE
     ]e[SPACE]r[SPACE]![SPACE]!"
210  h$=str$(h):f$="highscore:":f$=f$+h
     $:char0,15,11,f$
220  s$=str$(s):ifs>hthenf$="new[SPACE]
     highscore:":f$=f$+s$:char0,12,13,f
     $:h=s
230  char0,3,15,"press[SPACE]fire[SPACE
     ]button[SPACE]for[SPACE]another[SP
     ACE]game":do:loopuntiljoy(2)=128
240  goto10
250  color0,1:color4,1:scnclr:f=0:t=5:s
     =0:l=5:r=0:forx=3648to3711:pokex,0
     :next
260  poke3682,224:poke3684,1:poke3685,2
     40:poke3687,1:poke3688,240:poke369
     0,1
270  poke3691,240:poke3694,224:forx=358
     4to3647:ready:pokex,y:next
280  forx=0to39:print"[COM-8]-";:next:p
     rint"[CTRL-6]U";:forx=0to37:print"
     C";:next:print"I"
290  print"B[COM-6][SPACE]score:[SPACE]
     0[6xSPACE]level[SPACE]:[SPACE]1[5x
     SPACE]tries:[SPACE]5[CTRL-6][SPACE
     ]B"
300  print"J";:forx=0to37:print"[CTRL-6
     ]C";:next:print"K"
310  forx=0to39:print"[COM-8]-";:nextx
320  print"[CTRL-6][COM-D][17xCOM-I][3x
     SPACE][18xCOM-I][COM-F]"
330  forx=0to14:print"[CTRL-9][COM-K][C
     TRL 0]";spc(38);"[COM-K]":next:pri
     nt"[CRSR-DOWN]":forx=0to39:print"[
     CTRL-6][COM-R]";:next
340  movespr2,171,83:sprite2,1,2:movesp
     r1,171,216:sprite1,1,16:slow:retur
     n
350  data0,0,0,0,0,0,0,0,0,24,198,48,61
     ,239,120,61,239,120,24,198,48,0,0,
     0
360  data99,24,198,247,189,239,247,189,
     239,99,24,198,0,0,0,255,255,255,25
     5,255
370  data255,193,255,131,156,255,57,190
     ,255,125,54,0,108,62,0,124,28,0,56
     ,0
```

** EINDE LISTING catch-it

We staan zo langzamerhand weer voor een zomerperiode waarin traditioneel de meeste mensen meer vrije tijd hebben. Een deel van deze tijd zal dan ook wel weer achter de computer worden doorgebracht. Om deze tijd nuttig te besteden hebben we weer een groot aantal listings bij elkaar gezocht. Niet alleen spelletjes maar ook zijn er programma's bij welke u in uw eigen programma kunt gebruiken. Elders in het blad vindt u de uitslag van onze programeerwedstrijd waarvoor vele programma's zijn binnengekomen. Hierdoor zijn we weer in staat deze rubriek vele maanden te vullen.

R.Goudriaan

Syntax Checksum

Het overtikken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker om de fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.

Hiervoor gaat u als volgt te werk:

1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of cassette.

2. U tikt het RUN commando in. Mocht het programma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtikken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...'komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW commando geeft blijft het toch in de computer staan.

Alle door ons geplaatste programma's zijn in Basic geschreven.

Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden(wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4)in.

Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn. (Maandag 17.00 - 21.00 uur. Telefoonnummer 02155-25162.)

```
1     rem ********************************
      ***
2     rem basic loader "SYNTAX.CHECKSUM"
3     rem na de commando's "run" en "new
      "
4     rem blijft dit programma in het ge
      -
5     rem heugen. laad het te testen pro
      -
6     rem gramma en tik daarna sys 49152

7     rem ********************************
      ***
10    i=49152  :rem beginadres
20    reada:ifa<0then40:rem data ingelez
      en
30    pokei,a:i=i+1:b=b+a:goto20
40    if b<>16844thenprint"[SHIFT-CLR]fo
      ut[SPACE]in[SPACE]dataregels!":b=0
      :end
50    poke49184,148:poke49185,192
55    i=49300
60    read a: ifa<0then80
70    pokei,a:b=a+b:i=i+1:goto60
80    if b<>20068thenprint"[SHIFT-CLR]fo
      ut[SPACE]in[SPACE]dataregels![SPAC
      E](vanaf[SPACE]regel[SPACE]240)":b
      =0:end
90    print"data[SPACE]is[SPACE]weggezet
      "
95    print"checksum[SPACE]testen[SPACE]
      met[SPACE]sys49152"
100   data 165,43,166,44,133,163,134,164
      ,169,147
110   data 32,210,255,160,0,240,3,32,73,
      192
120   data 32,73,192,208,1,96,32,225,255
      ,208
130   data 3,76,116,164,32,81,192,32,73,
      192
140   data 240,12,201,32,240,247,24,101,
      167,133
150   data 167,76,37,192,166,167,169,0,1
      32,168
160   data 32,205,189,169,13,32,210,255,
      164,168
170   data 76,17,192,200,208,2,230,164,1
      77,163
180   data 96,162,0,189,123,192,240,6,32
      ,210
190   data 255,232,208,245,32,73,192,170
      ,32,73
200   data 192,132,168,32,205,189,162,3,
      169,32
210   data 32,210,255,202,208,250,169,0,
      133,167
220   data 164,168,96,82,69,71,69,76,32,
      0
230   data  -1
240   data 165,197,201,3,240,7,201,4,240
250   data 6,76,148,192,76,34,192,169
260   data 147,32,210,255,76,161,192
270   data -1
```

*** EINDE LISTING checksum 64 **

REGEL 1	249	REGEL 100	183
REGEL 2	84	REGEL 110	158
REGEL 3	105	REGEL 120	232
REGEL 4	2	REGEL 130	183
REGEL 5	246	REGEL 140	96
REGEL 6	152	REGEL 150	96
REGEL 7	249	REGEL 160	127
REGEL 10	157	REGEL 170	71
REGEL 20	64	REGEL 180	223
REGEL 30	38	REGEL 190	73
REGEL 40	57	REGEL 200	79
REGEL 50	14	REGEL 210	109
REGEL 55	251	REGEL 220	106
REGEL 60	192	REGEL 230	225
REGEL 70	42	REGEL 240	16
REGEL 80	244	REGEL 250	163
REGEL 90	245	REGEL 260	92
REGEL 95	237	REGEL 270	22

Checksum C 16

```
10      rem ******************************
        ***
20      rem syntax.checksum
30      rem voor c-16 & plus/4
40      rem
50      rem syntax testen met 'sys 1536'
60      rem
70      rem v.851128.16         jan bodzinga
80      rem ******************************
        ***
90      i=1536        :rem beginadres
100     reada:ifa>=0then pokei,a:i=i+1:got
        o100
110     print"data[SPACE]is[SPACE]weggezet
        "
120     print"cheksum[SPACE]printen[SPACE]
        met[SPACE]'sys[SPACE]1536'
130     end
200     data 165, 43,166, 44,133
210     data  31,134, 32,169,147
220     data  32,210,255,160,  0
230     data 240,  3, 32, 73,  6
240     data  32, 73,  6,208,  1
250     data  96, 72,152, 32,131
260     data   6,168,104,234, 32
270     data  81,  6, 32, 73,  6
280     data 240, 12,201, 32,240
290     data 247, 24,101,252,133
300     data 252, 76, 37,  6,166
310     data 252,169,  0,132,253
320     data  32, 95,164,169, 13
330     data  32,210,255,164,253
340     data  76, 17,  6,200,208
350     data   2,230, 32,177, 31
360     data  96,162,  0,189,123
370     data   6,240,  6, 32,210
380     data 255,232,208,245, 32
390     data  73,  6,170, 32, 73
400     data   6,132,253, 32, 95
410     data 164,162,  3,169, 32
420     data  32,210,255,202,208
430     data 250,169,  0,133,252
440     data 164,253, 96, 82, 69
450     data  71, 69, 76, 32,  0
460     data   0, 72,138, 72, 32
470     data 225,255,240,251,104
480     data 170,104, 96, -1
```

** EINDE LISTING checks16

REGEL 10	249	REGEL 280	186
REGEL 20	247	REGEL 290	248
REGEL 30	121	REGEL 300	118
REGEL 40	143	REGEL 310	204
REGEL 50	75	REGEL 320	165
REGEL 60	143	REGEL 330	252
REGEL 70	8	REGEL 340	106
REGEL 80	249	REGEL 350	98
REGEL 90	103	REGEL 360	163
REGEL 100	2	REGEL 370	45
REGEL 110	245	REGEL 380	0
REGEL 120	237	REGEL 390	58
REGEL 130	128	REGEL 400	108
REGEL 200	210	REGEL 410	159
REGEL 210	208	REGEL 420	245
REGEL 220	142	REGEL 430	202
REGEL 230	1	REGEL 440	176
REGEL 240	3	REGEL 450	12
REGEL 250	157	REGEL 460	54
REGEL 260	155	REGEL 470	43
REGEL 270	215	REGEL 480	1

Powertoetsen C 64

Met deze utilitie kan het aantal Funktie-toetsen in combinatie met de Powercartridge met 8 worden uitgebreid tot 16. Het grote voordeel is dat deze funkties herprogrammeerbaar zijn, het instellen gebeurt via een menu. De inzender van dit programma is R. Wolswijk.

```
10      rem power f-toetsen / commodore 64
20      rem
30      rem door r. wolswijk
40      rem
50      rem *** f9 t/m f16 als aanv. op de
         powercart. ***
60      dim f$(20):print"[SHIFT-CLR]";:for
         k=9 to 16:read f$(k):next
70      t1=51968:t2=52195:for k=t1 to t2:r
        ead a:poke k,a:next
80      gosub 220:gosub 200
90      p$="":t$="":input"[CRSR-UP][CTRL-9
        ][SPACE]welke[SPACE]toets[SPACE](9
        -16)[CTRL-0][15xSPACE][15xCRSR-LEF
        T]";t$
100     t=val(t$):if t=0then 80
110     if t<9 or t>16 then 90
120     ff$="":print"[CRSR-UP][CTRL-9][SPA
        CE]welke[SPACE]functie[SPACE]f";t;
        "[CTRL-0][CRSR-LEFT][9xSPACE][8xCR
        SR-LEFT]";:input ff$
130     if ff$="" then gosub200:goto 90
140     l=len(ff$):if l>10then gosub 220:
        goto 120
150     if right$(ff$,1)="[pijl[SPACE]link
        s]" then l=l-1:p$="[plijl[SPACE]li
        nks]"
160     f$(t)=ff$:pl=52008+t*11:for p=1 to
         l:poke pl+p,asc(mid$(f$(t),p,1)):
        next
170     if p$="[pijl[SPACE]links]" then po
        kepl+p,13:p=p+1
180     if p>10then 80
190     for k=p to 10:poke pl+k,0:next:got
        o 80
200     a$="":input"[CTRL-9][CRSR-UP][SPAC
        E]nog[SPACE]func.toetsen[SPACE]wij
        zigen(j/n)[CTRL-0]";a$:if a$="j" t
        hen return
210     sys t1:new
220     print"[SHIFT-CLR]";:for p=9 to 16:
        print"f";right$(str$(p),2);"[SPACE
```

```
         ]";f$(p)
230      print:next:print"[CTRL-9]restart:s
         ys[SPACE]51968[2xCRSR-DOWN]":retur
         n
240      data"coL1,1,0[pijl[SPACE]links]","
         coL0,0,1[pijl[SPACE]links]",dload"
         ,"repeat[pijl links]"
250      data"print","tab(","gosub","data"
260      rem *** machine-codes ***
270      data 120, 169, 18, 141, 20, 3, 169
         , 203, 141, 21, 3, 88, 169
280      data 203, 133, 254, 96, 234, 174,
         141, 2, 165, 197, 201, 4, 240
290      data 19, 201, 5, 240, 33, 201, 6,
         240, 47, 201, 3, 240, 61
300      data 169, 0, 133, 2, 76, 49, 234,
         165, 2, 208, 249, 224, 2
310      data 240, 6, 144, 243, 169, 184, 2
         08, 56, 169, 140, 208, 52, 165
320      data 2, 208, 231, 224, 2, 240, 6,
         144, 225, 169, 195, 208, 38
330      data 169, 151, 208, 34, 165, 2, 20
         8, 213, 224, 2, 240, 6, 144
340      data 207, 169, 206, 208, 20, 169,
         162, 208, 16, 165, 2, 208, 195
350      data 224, 2, 240, 6, 144, 189, 169
         , 217, 208, 2, 169, 173, 133
360      data 253, 160, 0, 177, 253, 153, 1
         19, 2, 200, 192, 10, 208, 246
370      data 132, 198, 169, 255, 133, 2, 7
         6, 49, 234, 234, 67, 79, 204
380      data 49, 44, 49, 44, 48, 13, 0, 23
         4, 67, 79, 204, 48, 44
390      data 48, 44, 49, 13, 0, 234, 0, 0,
          0, 0, 68, 76, 79
400      data 65, 68, 34, 234, 82, 69, 80,
         69, 65, 84, 13, 0, 0
410      data 0, 234, 80, 82, 73, 78, 84, 0
         , 0, 0, 0, 0, 234
420      data 84, 65, 66, 40, 0, 0, 0, 0, 0
         , 0, 234, 71, 79
430      data 83, 85, 66, 0, 0, 0, 0, 0, 23
         4, 68, 65, 84, 65
440      data 0, 0, 0, 0, 0, 0, 234
```

** EINDE LISTING pftoetsen

REGEL 10	239	REGEL 280	96
REGEL 20	143	REGEL 290	78
REGEL 30	189	REGEL 300	112
REGEL 40	143	REGEL 310	165
REGEL 50	200	REGEL 320	1
REGEL 60	7	REGEL 330	248
REGEL 70	106	REGEL 340	161
REGEL 80	122	REGEL 350	60
REGEL 90	85	REGEL 360	88
REGEL 100	158	REGEL 370	79
REGEL 110	247	REGEL 380	29
REGEL 120	251	REGEL 390	12
REGEL 130	35	REGEL 400	183
REGEL 140	135	REGEL 410	252
REGEL 150	53	REGEL 420	203
REGEL 160	75	REGEL 430	16
REGEL 170	40	REGEL 440	68
REGEL 180	252		
REGEL 190	120		
REGEL 200	177		
REGEL 210	255		
REGEL 220	230		
REGEL 230	211		
REGEL 240	225		
REGEL 250	61		
REGEL 260	27		
REGEL 270	203		

Schermwissen C 64

Dit programmaatje door Marcel de Groot geschreven zet een machinecode-programmaatje weg, waardoor met 'sys 49152' het scherm op een leuke manier gewist wordt. Dit gaat alleen goed, wanneer er maar een kleur op het scherm staat. Dit programma is dus niet bedoeld als zelfstandig programma maar kan in uw eigen programma's worden gebruikt om dit een speelser karakter te geven.

```
10       fori=0to75:reada:poke49152+i,a:nex
         t:a=peek(646):fori=1024to2023:poke
         i,208
20       poke54272+i,a:next:print"[HOME]"sp
         c(17)"[5xCRSR-DOWN]toets!"chr$(142
          ):poke198,0:wait198,1
30        sys49152:data120,162,45,160,255,16
          9,27,141,17,208,173,33,208,141,60,
         192,32,47
40       data192,232,136,192,150,208,247,88
          ,169,147,32,210,255,120,32,47,192,
         202,200
50       data208,249,173,60,192,141,33,208,
         88,96,169,27,205,17,208,208,251,23
         6,18,208
60        data208,251,169,0,141,33,208,204,1
         8,208,208,251,173,134,2,141,33,208
         ,96
```

** EINDE LISTING schermwissen

REGEL 10	88
REGEL 20	170
REGEL 30	184
REGEL 40	54
REGEL 50	132
REGEL 60	10

Super Screen C 64

Super screen is bedoeld om het programmeren, nakijken en intypen wat makkelijker te maken Als de data's ingelezen zijn en de opdracht 'sys 49152' is gegeven, wordt het scherm grijs en er verschijnt een donker grijze balk over de plaats waar de cursor staat. De balk maakt het de gebruiker makkelijker om zich te concentreren Elke keer als men een toets aanslaat hoort men een toon als men nu tweemaal een toets indrukt in plaats van één dan is dit ook te horen Ook kent dit programma een zogenaamd klad blad. in regel 30 staat poke 49404,57 - 57 is de code voor de toets [pijl links] Een ander getal (b.v 4) geeft een andere toets (b.v. F1) Met deze toets is het mogelijk om te schakelen van scherm 1 (gewoon) naar scherm 2. (klad blad). Als u dit de eerste maal doet, staat er allemaal onzin op het klad blad, maar deze is na [Shift-clr] verdwenen. Als de gebruiker op het ene scherm werkt, blijft alles gewoon op het andere scherm staan. Dit heeft grote voordelen. U kunt werken op het ene scherm en op het andere scherm dingen zetten die men wil onthouden, of bewaren. Super screen uit zetten gebeurt door terug te gaan naar het gewone scherm en niet klad blad (belangrijk) en op RUN/Stop/Restore te drukken. Weer aan gaat met sys 49152 Superscreen zal wel werken met de meeste basic programma's maar niet met de meeste ma-

chinetaal programma's. Super-screen gebruikt de geheugenruimte van 49152 tot 53248 De inzender van dit programma is Abbe Vermey. en wij hopen met hem dat u veel aan dit programma hebt.

```
0      rem **********super-screen********
       **
1      rem
2      rem    *************************
3      rem   **     obbe      vermeij    **
4      rem  **                           **
5      rem  **      capelle a/d ij.      **
6      rem   **                         **
7      rem    *************************
8      rem
9      rem
10     print "[4xCRSR-RIGHT][6xCRSR-DOWN]
       [CTRL-2]enkele[SPACE]seconden[SPAC
       E]geduld[SPACE]a.u.b"
20     x=0:for t=49152 to 49652 : read a
       : poke t , a :x = x + a : next
25     if x <> 62909 then print "[CTRL-1]
       [SPACE]fout[SPACE]in[SPACE]data[SP
       ACE]!![SPACE]":end
30     poke 49404,57
40     sys 49152
50     end
100    data 173,0,193,240,0,173,14,220,41
       ,254,141,14,220,165,1,41,251,133,1
       ,162,0
110    data 134,251,162,208,134,252,160,0
       ,177,251,168,216,56,165,252,237,0,
       193,133
120    data252,152,160,0,145,251,230,251,
       208,231,232,224,212,208,226,173,0,
       193,201
130    data 4,240,8,169,4,141,0,193,76,19
       ,192,169,0,141,0,193,165,1,9,4,133
       ,1,173
140    data 14,220,9,1,141,14,220,169,203
       ,133,252,198,251,216,24,165,251,10
       5,8,133
150    data 251,201,7,240,9,169,255,81,25
       1,145,251,76,95,192,230,252,165,25
       2,201
160    data 208,208,239,169,255,141,18,20
       8,173,17,208,41,127,141,17,208,120
       ,169
170    data 129,141,26,208,169,192,141,21
       ,3,169,163,141,20,3,169,0,141,255,
       194,88
180    data 32,226,193,96,173,25,208,141,
       25,208,48,7,173,13,220,88,76,1,193
       ,173
190    data 255,194,208,26,238,255,194,16
       9,12,141,33,208,169,11,141,32,208,
       216,24
200    data 173,18,208,105,11,141,18,208,
       76,188,254,206,255,194,169,15,141,
       33,208
210    data 169,15,141,32,208,169,80,141,
       18,208,56,32,240,255,138,10,10,10,
       141,254
220    data 194,169,48,109,254,194,141,18
       ,208,76,188,254,0,1,1,57,193,76,18
       8,8,165
230    data 197,201,64,208,8,169,0,141,24
       9,192,76,163,193,173,249,192,240,7
       ,173
240    data 250,192,197,197,240,241,165,1
       97,141,250,192,141,249,192,169,64,
       141,14
250    data 212,141,15,212,169,0,141,19,2
       12,169,51,141,20,212,169,15,141,24
       ,212
260    data 169,64,141,18,212,169,65,141,
       18,212,165,197,205,252,192,240,3,7
       6,168
270    data 193,173,251,192,240,39,169,0,
       141,251,192,173,2,221,9,3,141,2,22
       1,173
280    data 0,221,41,252,141,0,221,173,24
       ,208,41,240,9,2,141,24,208,169,196
       ,141
290    data 136,2,76,49,234,173,2,221,9,3
       ,141,2,221,173,0,221,41,252,9,3,14
       1,0,221
300    data 173,24,208,41,240,9,4,141,24,
       208,169,4,141,136,2,169,1,141,251,
       192,76
310    data 49,234,169,0,141,24,212,173,2
       51,192,208,243,162,0,189,194,193,1
       57,4
320    data 196,169,0,157,4,216,232,224,3
       2,240,41,76,175,193,32,15,3,20,15,
       16,21
330    data 19,32,32,19,15,6,20,23,1,18,5
       ,32,32,32,32,11,12,1,4,45,2,12,1,4
       ,32,169
340    data 1,141,134,2,96,173,24,208,41,
       15,9,16,141,24,208,76,49,234
```

** EINDE LISTING s-screen

REGEL 0	83	REGEL 150	168
REGEL 1	143	REGEL 160	125
REGEL 2	169	REGEL 170	216
REGEL 3	97	REGEL 180	129
REGEL 4	55	REGEL 190	221
REGEL 5	162	REGEL 200	225
REGEL 6	55	REGEL 210	245
REGEL 7	169	REGEL 220	47
REGEL 8	143	REGEL 230	142
REGEL 9	143	REGEL 240	237
REGEL 10	168	REGEL 250	80
REGEL 20	133	REGEL 260	182
REGEL 25	80	REGEL 270	149
REGEL 30	52	REGEL 280	82
REGEL 40	163	REGEL 290	218
REGEL 50	128	REGEL 300	202
REGEL 100	159	REGEL 310	123
REGEL 110	5	REGEL 320	147
REGEL 120	21	REGEL 330	197
REGEL 130	199	REGEL 340	12
REGEL 140	240		

Joystick prog C 64

Dit programma is een hulpprogramma om sprites aan één van de joysticks te koppelen. Een uitleg is in het programma opgenomen. De inzender is M klabbers uit Son.

```
10     gosub270:print"[SHIFT-CLR][2xCRSR-
       DOWN][SPACE]even[SPACE]denken..."
20     fora=0to483:readq:poke49248+a,q:ck
       =ck+q:next:ifck=66230then40
30     print"[CRSR-DOWN][SPACE]foutje":en
       d
40     rem
50     rem  martijn klabbers
60     rem
70     rem        son
80     rem
```

```
90    print"[SHIFT-CLR][2xCRSR-DOWN][SPA
      CE]de[SPACE]data[SPACE]zijn[SPACE]
      opgeslagen.
100   print"[CRSR-DOWN][SPACE]wilt[SPACE
      ]u[SPACE]ze[SPACE]no[SPACE]als[SPA
      CE]m.l.[SPACE]wegsaven[SPACE]?"
110   poke198,0:wait198,1:geta$:ifa$="n"
      then end
120   end
130   end
140   end
150   print"[SHIFT-CLR][2xCRSR-DOWN][SPA
      CE]waar[SPACE]wilt[SPACE]u[SPACE]h
      et[SPACE]op[SPACE]wegsaven[SPACE]?
      "
160   print"[CRSR-DOWN][SPACE]([CTRL-9]c
      [CTRL-0]assette/[CTRL-9]d[CTRL-0]i
      skette)":d=8
170   poke198,0:wait198,1:geta$:ifa$="c"
      thend=1
180   input"[CRSR-DOWN][SPACE]en[SPACE]d
      e[SPACE]naam[SPACE]";a$
190   a$=a$+"[15xSPACE]":fora=1to16:poke
      49191+a,asc(mid$(a$,a,1)):next
200   fora=0to34:readq:poke49152+a,q:kc=
      kc+q:next:poke49153,d
210   ifkc<>5194thenprint"[CRSR-DOWN][SP
      ACE]foutje":end
220   print"[CRSR-DOWN][SPACE]klaarmaken
      ....";:poke204,0:poke198,0:wait198
      ,1:poke204,1
230   sys49152
240   print:print"[CRSR-DOWN][SPACE]klaa
      r."
250   end
260   rem joystickprogramma data
270   rem
280   data141,095,195,169,001,133,254,17
      3
290   data064,192,170,189,075,192,134,25
      5
300   data168,010,170,169,000,200,056,04
      2
310   data136,208,252,168,132,253,173,09
      5
320   data195,074,072,176,026,173,091,19
      5
330   data208,021,189,001,208,205,067,19
      2
340   data208,010,165,254,240,006,152,14
      1
350   data091,195,208,003,222,001,208,10
      4
360   data074,072,176,026,173,093,195,20
      8
370   data021,189,001,208,205,070,192,20
      8
380   data010,165,254,240,006,152,141,09
      3
390   data195,208,003,254,001,208,104,07
      4
400   data072,176,050,173,090,195,208,04
      5
410   data189,000,208,205,065,192,208,02
      0
420   data165,254,240,016,173,066,192,07
      7
430   data016,208,037,253,208,006,152,14
      1
440   data090,195,208,017,222,000,208,18
      9
450   data000,208,201,255,208,007,152,07
      7
460   data016,208,141,016,208,104,074,07
      2
470   data176,048,173,092,195,208,043,18
      9
480   data000,208,205,068,192,208,020,16
      5
490   data254,240,016,173,069,192,077,01
      6
500   data208,037,253,208,006,152,141,09
      2
510   data195,208,015,254,000,208,189,00
      0
520   data208,208,007,152,077,016,208,14
      1
530   data016,208,104,074,176,004,169,00
      1
540   data208,002,169,000,141,094,195,16
      9
550   data000,133,254,166,255,202,240,00
      3
560   data076,107,192,096,096,096,096,09
      6
570   data120,169,049,162,234,141,020,00
      3
580   data142,021,003,088,096,096,096,09
      6
590   data096,096,096,096,032,072,193,16
      0
600   data001,132,253,032,253,174,032,15
      8
610   data183,224,000,240,077,224,021,14
      4
620   data005,162,014,076,055,164,142,00
      0
630   data192,134,255,032,253,174,032,15
      8
640   data183,224,008,176,236,138,166,25
      5
650   data157,011,192,202,208,235,160,00
      0
660   data132,254,032,253,174,032,235,18
      3
670   data138,164,254,153,003,192,165,02
      0
680   data153,001,192,165,021,240,002,16
      5
690   data255,153,002,192,192,003,240,00
      4
700   data160,003,208,220,032,253,174,03
      2
710   data158,183,165,253,240,021,134,25
      2
720   data142,007,192,162,032,189,255,19
      1
730   data157,031,192,202,208,247,134,25
      3
740   data076,099,193,134,251,142,007,19
      2
750   data120,169,193,162,229,141,021,00
      3
760   data142,020,003,088,096,169,000,16
      2
770   data016,157,079,195,202,208,250,16
      5
780   data251,240,037,198,251,208,033,17
      3
790   data007,192,133,251,162,032,189,25
      5
800   data191,157,063,192,202,208,247,17
```

```
     3
810  data001,220,032,096,192,162,005,18
     9
820  data089,195,157,079,195,202,208,24
     7
830  data165,252,240,037,198,252,208,03
     3
840  data173,039,192,133,252,162,032,18
     9
850  data031,192,157,063,192,202,208,24
     7
860  data173,000,220,032,096,192,162,00
     5
870  data189,089,195,157,084,195,202,20
     8
880  data247,076,049,234
890  rem
900  rem data m.l.wegsaven
910  rem
920  data 162,8,169,1,168,32,186,255,16
     9,16,162,40,160,192,32,189,255,169
     ,96
930  data 133,251,169,192,133,252,169,2
     51,162,68,160,194,32,216,255,96
940  return
950  rem      joystickprogramma
960  rem dit programma stelt u in
970  rem staat om een of meerdere
980  rem sprites te koppelen aan een
990  rem of allebei de joysticks
1000 rem
1010 rem      gebruiksaanwijzing
1020 rem
1030 rem u roept het programma aan
1040 rem met :'sys 49500,....'
1050 rem (achter dit commando volgen
1060 rem nog enkele parameters)
1070 rem het programma loopt via het
1080 rem interrupt zodat u er geen
1090 rem last van heeft tijdens een
1100 rem programma.
1110 rem compleet ziet de opdracht er
1120 rem zo uit:
1130 rem'sys49500,a,sl,enz,x,y,p,q,sp
1140 rem (en daarna het zelfde voor
1150 rem de andere joystick).'
1160 rem
1170 rem-achter de eerste komma staat
1180 rem hoeveel sprites u aan de
1190 rem joystick in 'port 2' wilt
1200 rem koppelen gevolgd door de
1210 rem spritenummers ( 0t/m 7 )
1220 rem dit getal ('a') mag ook
1230 rem groter zijn dan '8'     ,,
1240 rem u koppelt in dat geval een
1250 rem sprite twee of meer malen
1260 rem aan de joystick waardoor
1270 rem die sprite twee of meer
1280 rem malen sneller gaat.
1290 rem
1300 rem-achter de spritenummers
1310 rem komen de x en y-coordinaten
1320 rem van de linker bovenhoek
1330 rem ( x en y) en de rechter
1340 rem onderhoek ( p en q ) van het
1350 rem blok waarin de sprite zich
1360 rem mag bewegen.
1370 rem
1380 rem-daarachter komt de snelheid
1390 rem waarmee de sprites zich
1400 rem mogen bewegen.deze loopt
1410 rem van 1 (snelst) tot hoger
1420 rem (langzamer).met de waarde'0'
1430 rem gaan de sprites het lang-
1440 rem zaamst.
1450 rem
1460 rem daarna volgt hetzelfde voor
1470 rem de joystick in 'port 1'
1480 rem
1490 rem wilt u dat een van de
1500 rem joysticks niet wordt
1510 rem ingeschakelt dan neemt u
1520 rem voor het sprite-aantal van
1530 rem de betreffende joystick het
1540 rem getal '0';de andere para-
1550 rem meters moeten weggelaten
1560 rem worden voor die betreffende
1570 rem joystick.
1580 rem
1590 rem het interrupt schakelt u uit
1600 rem met het commando:'sys49480'.
1610 rem het interrupt volgt dan
1620 rem zijn normale weg.
1630 rem ook met 'stop' en 'restore'
1640 rem wordt ditzelfde bereikt.
1650 rem
1660 rem u kunt ook de sprite die
1670 rem het eerst na het sprite-
1680 rem aantal komt,controleren of
1690 rem hij niet tegen het vastge-
1700 rem stelde blok komt.
1710 rem dit kan met de registers:
1720 rem 50000- 50003 (port 1)
1730 rem 50005 - 50008 (port 2)
1740 rem ( 50000/5 voor 'x' )
1750 rem ( 50001/6 voor 'y' )
1760 rem ( 50002/7 voor 'p' )
1770 rem ( 50003/8 voor 'q' )
1780 rem hetzelfde geldt voor'port 2'
1790 rem
1800 rem de vuurknop wordt gedetec-
1810 rem teerd in 50004 voor 'port 1'
1820 rem en in 50009 voor 'port 2'.
1830 rem
1840 rem u kunt dit programma in uw
1850 rem eigen programma verwerken
1860 rem door de data-regels in het
1870 rem programma te verwerken.
1880 rem u kunt ook ,en dat is het
1890 rem beste, het wegge'save'de
1900 rem programma in uw programma
1910 rem laden met de regel:
1920 rem 0ifh=0thenh=1:load"naam",8,1
1930 rem of voor cassette:
1940 rem 0if h=0then h=1 :load"naam"
1950 rem
1960 rem                 succes !
1970 rem                    (m.klabbers)
```

** EINDE LISTING joystick

REGEL 10	95	REGEL 130	128
REGEL 20	235	REGEL 140	128
REGEL 30	117	REGEL 150	113
REGEL 40	143	REGEL 160	215
REGEL 50	234	REGEL 170	209
REGEL 60	143	REGEL 180	179
REGEL 70	127	REGEL 190	97
REGEL 80	143	REGEL 200	5
REGEL 90	97	REGEL 210	108
REGEL 100	51	REGEL 220	217
REGEL 110	53	REGEL 230	163
REGEL 120	128	REGEL 240	90

REGEL 250	128	REGEL 600	118	REGEL 950	165	REGEL 1300	250	REGEL 1650	143
REGEL 260	191	REGEL 610	115	REGEL 960	142	REGEL 1310	57	REGEL 1660	65
REGEL 270	143	REGEL 620	115	REGEL 970	94	REGEL 1320	99	REGEL 1670	103
REGEL 280	136	REGEL 630	133	REGEL 980	88	REGEL 1330	77	REGEL 1680	135
REGEL 290	147	REGEL 640	145	REGEL 990	94	REGEL 1340	121	REGEL 1690	229
REGEL 300	114	REGEL 650	113	REGEL 1000	143	REGEL 1350	7	REGEL 1700	225
REGEL 310	140	REGEL 660	126	REGEL 1010	221	REGEL 1360	143	REGEL 1710	171
REGEL 320	148	REGEL 670	127	REGEL 1020	143	REGEL 1370	143	REGEL 1720	112
REGEL 330	129	REGEL 680	112	REGEL 1030	197	REGEL 1380	155	REGEL 1730	123
REGEL 340	115	REGEL 690	115	REGEL 1040	226	REGEL 1390	114	REGEL 1740	37
REGEL 350	115	REGEL 700	111	REGEL 1050	104	REGEL 1400	230	REGEL 1750	40
REGEL 360	145	REGEL 710	129	REGEL 1060	68	REGEL 1410	59	REGEL 1760	33
REGEL 370	123	REGEL 720	136	REGEL 1070	101	REGEL 1420	80	REGEL 1770	36
REGEL 380	117	REGEL 730	129	REGEL 1080	233	REGEL 1430	137	REGEL 1780	165
REGEL 390	121	REGEL 740	141	REGEL 1090	253	REGEL 1440	141	REGEL 1790	143
REGEL 400	137	REGEL 750	121	REGEL 1100	99	REGEL 1450	143	REGEL 1800	80
REGEL 410	125	REGEL 760	123	REGEL 1110	153	REGEL 1460	163	REGEL 1810	157
REGEL 420	140	REGEL 770	139	REGEL 1120	100	REGEL 1470	227	REGEL 1820	240
REGEL 430	122	REGEL 780	132	REGEL 1130	190	REGEL 1480	143	REGEL 1830	143
REGEL 440	131	REGEL 790	133	REGEL 1140	210	REGEL 1490	67	REGEL 1840	240
REGEL 450	119	REGEL 800	138	REGEL 1150	181	REGEL 1500	18	REGEL 1850	86
REGEL 460	120	REGEL 810	124	REGEL 1160	143	REGEL 1510	162	REGEL 1860	205
REGEL 470	153	REGEL 820	158	REGEL 1170	182	REGEL 1520	80	REGEL 1870	181
REGEL 480	122	REGEL 830	135	REGEL 1180	111	REGEL 1530	131	REGEL 1880	36
REGEL 490	138	REGEL 840	138	REGEL 1190	155	REGEL 1540	62	REGEL 1890	132
REGEL 500	126	REGEL 850	132	REGEL 1200	178	REGEL 1550	10	REGEL 1900	30
REGEL 510	125	REGEL 860	116	REGEL 1210	21	REGEL 1560	144	REGEL 1910	11
REGEL 520	127	REGEL 870	159	REGEL 1220	123	REGEL 1570	45	REGEL 1920	100
REGEL 530	123	REGEL 880	119	REGEL 1230	78	REGEL 1580	143	REGEL 1930	0
REGEL 540	133	REGEL 890	143	REGEL 1240	186	REGEL 1590	211	REGEL 1940	163
REGEL 550	111	REGEL 900	254	REGEL 1250	198	REGEL 1600	98	REGEL 1950	143
REGEL 560	163	REGEL 910	143	REGEL 1260	183	REGEL 1610	156	REGEL 1960	118
REGEL 570	116	REGEL 920	101	REGEL 1270	43	REGEL 1620	233	REGEL 1970	161
REGEL 580	144	REGEL 930	47	REGEL 1280	92	REGEL 1630	247		
REGEL 590	149	REGEL 940	142	REGEL 1290	143	REGEL 1640	238		

Digitale klok C 64

Klokken, het blijft de inzenders en hopelijk ook de lezers boeien.Ditmaal staat er in grote cijfers de door u aangegeven tijd op het scherm. Dit programma klok werd ons toegezonden door Eric Goldstein.

```
10    rem ******************************
      **
20    rem *         digitale klok
       *
30    rem ******************************
      **
40    ta=0
50    gosub 1490
60    poke 53280,0:poke53281,2
70    input"[SHIFT-CLR][CTRL-8][5xCRSR-D
      OWN][3xCRSR-RIGHT]geef[SPACE]de[SP
      ACE]tijd[SPACE](uummss):";ta$
80    iflen(ta$)>=6thenti$=ta$
90    v=53248
100   fori=0to7
110   pokev+i*2,(i*40+20)and 255
120   pokev+i*2+1,130
130   pokev+39+i,7
140   ifi=2ori=5thenpokev+39+i,0
150   poke2040+i,244
160   next
170   pokev+33,2
180   pokev+32,0
190   pokev+21,255
200   pokev+23,255
210   pokev+29,255
220   pokev+16,0
230   poke2042,255:poke2045,255
240   pokev+16,192
250   print"[CTRL-2][SHIFT-CLR][8xCRSR-D
      OWN][3xCRSR-RIGHT]uren[9xSPACE]min
      uten[8xSPACE]seconden"
260   fori=0to7
270   ifi=2ori=5then320
280   a=i+1:ifa>6thena=a-1
290   ifa>3thena=a-1
300   b=val(mid$(ti$,a,1))
310   poke2040+i,b+245
320   next
330   pokev+44,0:clr:t$=ti$:v=53248
340   ifti$=t$then340
350   pokev+44,2:goto260
360   rem *** data voor cijfers ***
370   data 246
380   data 0,0,0,0,0,0,0
390   data 0,0,0,0,0,0,28
400   data 0,0,36,0,0,68,0
410   data 0,132,0,1,4,0,1
420   data 100,0,1,164,0,0,36
430   data 0,0,36,0,0,36,0
440   data 0,36,0,0,36,0,0
450   data 36,0,0,36,0,1,231
460   data 128,1,0,128,1,255,128
470   data 247
480   data 0,0,0,0,0,0,0
490   data 0,0,0,0,0,0,126
500   data 0,0,129,0,1,60,128
510   data 2,66,64,2,66,64,2
520   data 66,64,3,194,64,0,4
530   data 128,0,9,0,0,18,0
540   data 0,36,0,0,72,0,0
550   data 144,0,1,32,0,2,127
560   data 192,2,0,64,3,255,192
570   data 248
580   data 0,0,0,0,0,0,0
590   data 0,0,0,0,0,0,124
600   data 0,0,130,0,1,57,0
610   data 2,68,128,2,68,128,3
620   data 196,128,0,4,128,0,9
630   data 0,0,18,0,0,9,0
640   data 0,4,128,3,196,128,2
650   data 68,128,2,68,128,1,57
```

```
660   data 0,0,130,0,0,124,0
670   data 249
680   data 0,0,0,0,0,0,0
690   data 0,0,0,0,0,0,15
700   data 0,0,17,0,0,33,0
710   data 0,73,0,0,153,0,1
720   data 41,0,2,73,0,4,137
730   data 0,4,249,224,4,0,32
740   data 7,249,224,0,9,0,0
750   data 9,0,0,9,0,0,9
760   data 0,0,9,0,0,15,0
770   data 250
780   data 0,0,0,0,0,0,0
790   data 0,0,0,0,0,3,255
800   data 192,2,0,64,2,127,192
810   data 2,64,0,2,64,0,2
820   data 64,0,2,126,0,2,1
830   data 0,3,252,128,0,2,64
840   data 0,2,64,0,2,64,3
850   data 194,64,2,66,64,1,60
860   data 128,0,129,0,0,126,0
870   data 251
880   data 0,0,0,0,0,0,0
890   data 0,0,0,0,0,0,127
900   data 0,0,128,128,1,62,64
910   data 2,65,192,2,64,0,2
920   data 64,0,2,64,0,2,126
930   data 0,2,1,0,2,60,128
940   data 2,66,64,2,66,64,2
950   data 66,64,2,66,64,1,60
960   data 128,0,129,0,0,126,0
970   data 252
980   data 0,0,0,0,0,0,0
990   data 0,0,0,0,0,3,255
1000  data 192,2,0,64,3,254,64
1010  data 0,2,64,0,2,64,0
1020  data 4,128,0,4,128,0,9
1030  data 0,0,9,0,0,18,0
1040  data 0,18,0,0,36,0,0
1050  data 36,0,0,72,0,0,72
1060  data 0,0,144,0,0,240,0
1070  data 253
1080  data 0,0,0,0,0,0,0
1090  data 0,0,0,0,0,0,124
1100  data 0,0,130,0,1,57,0
1110  data 2,68,128,2,68,128,2
1120  data 68,128,2,68,128,1,57
1130  data 0,0,130,0,1,57,0
1140  data 2,68,128,2,68,128,2
1150  data 68,128,2,68,128,1,57
1160  data 0,0,130,0,0,124,0
1170  data 254
1180  data 0,0,0,0,0,0,0
1190  data 0,0,0,0,0,0,126
1200  data 0,0,129,0,1,60,128
1210  data 2,66,64,2,66,64,2
1220  data 66,64,2,66,64,1,60
1230  data 64,0,128,64,0,126,64
1240  data 0,2,64,0,2,64,0
1250  data 2,64,3,130,64,2,124
1260  data 128,1,1,0,0,254,0
1270  data 245
1280  data 0,0,0,0,0,0,0
1290  data 0,0,0,0,0,0,255
1300  data 0,1,0,128,2,126,64
1310  data 4,129,32,4,130,32,4
1320  data 132,32,4,137,32,4,147
1330  data 32,4,165,32,4,201,32
1340  data 4,145,32,4,33,32,4
1350  data 65,32,4,129,32,2,126
1360  data 64,1,0,128,0,255,0
1370  data 255
1380  data 0,0,0,0,0,0,0
1390  data 0,0,0,0,0,0,60
1400  data 0,0,126,0,0,126,0
1410  data 0,126,0,0,126,0,0
1420  data 60,0,0,0,0,0,0
1430  data 0,0,0,0,0,0,0
1440  data 0,0,0,0,60,0,0
1450  data 126,0,0,126,0,0,126
1460  data 0,0,126,0,0,60,0
1470  data -1
1480  rem *** sprite data lezen ***
1490  read a:ifa=-1then1540
1500  print"[SHIFT-CLR]sprite[SPACE]:";t
      a:ta=ta+1
1510  fori=0to62:readx
1520  poke64*a+i,x:next
1530  goto 1490
1540  fori=0to62:poke244*64+i,0:next
1550  return
```

** EINDE LISTING di-klok

REGEL 10	207	REGEL 270	212	REGEL 530	136	REGEL 790	74	REGEL 1050	134
REGEL 20	87	REGEL 280	187	REGEL 540	77	REGEL 800	91	REGEL 1060	170
REGEL 30	207	REGEL 290	103	REGEL 550	230	REGEL 810	85	REGEL 1070	29
REGEL 40	119	REGEL 300	176	REGEL 560	94	REGEL 820	131	REGEL 1080	219
REGEL 50	91	REGEL 310	3	REGEL 570	33	REGEL 830	238	REGEL 1090	66
REGEL 60	39	REGEL 320	130	REGEL 580	219	REGEL 840	86	REGEL 1100	124
REGEL 70	218	REGEL 330	209	REGEL 590	66	REGEL 850	50	REGEL 1110	51
REGEL 80	196	REGEL 340	180	REGEL 600	124	REGEL 860	27	REGEL 1120	108
REGEL 90	14	REGEL 350	184	REGEL 610	52	REGEL 870	27	REGEL 1130	124
REGEL 100	135	REGEL 360	241	REGEL 620	46	REGEL 880	219	REGEL 1140	51
REGEL 110	235	REGEL 370	31	REGEL 630	29	REGEL 890	69	REGEL 1150	108
REGEL 120	89	REGEL 380	219	REGEL 640	42	REGEL 900	36	REGEL 1160	166
REGEL 130	89	REGEL 52428	70	REGEL 650	108	REGEL 910	194	REGEL 1170	30
REGEL 140	145	REGEL 400	82	REGEL 660	166	REGEL 920	188	REGEL 1180	219
REGEL 150	22	REGEL 410	71	REGEL 670	34	REGEL 930	129	REGEL 1190	68
REGEL 160	130	REGEL 420	225	REGEL 680	219	REGEL 940	205	REGEL 1200	233
REGEL 170	91	REGEL 430	77	REGEL 690	17	REGEL 950	0	REGEL 1210	205
REGEL 180	88	REGEL 440	77	REGEL 700	73	REGEL 960	27	REGEL 1220	0
REGEL 190	194	REGEL 450	180	REGEL 710	127	REGEL 970	28	REGEL 1230	93
REGEL 200	196	REGEL 460	138	REGEL 720	187	REGEL 980	219	REGEL 1240	83
REGEL 210	202	REGEL 470	32	REGEL 730	239	REGEL 990	74	REGEL 1250	33
REGEL 220	90	REGEL 480	219	REGEL 740	194	REGEL 1000	43	REGEL 1260	179
REGEL 230	139	REGEL 490	68	REGEL 750	246	REGEL 1010	83	REGEL 1270	30
REGEL 240	198	REGEL 500	233	REGEL 760	26	REGEL 1020	194	REGEL 1280	219
REGEL 250	253	REGEL 510	205	REGEL 770	26	REGEL 1030	29	REGEL 1290	71
REGEL 260	135	REGEL 520	0	REGEL 780	219	REGEL 1040	77	REGEL 1300	236

REGEL 1310	33	REGEL 1440	17
REGEL 1320	138	REGEL 1450	22
REGEL 1330	81	REGEL 1460	122
REGEL 1340	241	REGEL 1470	225
REGEL 1350	91	REGEL 1480	250
REGEL 1360	237	REGEL 1490	205
REGEL 1370	31	REGEL 1500	66
REGEL 1380	219	REGEL 1510	209
REGEL 1390	17	REGEL 1520	33
REGEL 1400	173	REGEL 1530	87
REGEL 1410	173	REGEL 1540	68
REGEL 1420	17	REGEL 1550	140
REGEL 1430	219		

Lockin Man C 64

Ook dit programma is er één van de hand van Erik Goldstein. Hij heeft het programma Lockin Man genoemd. De naam verraad de bedoeling van het spel al. U moet de figuurtjes proberen in het doolhof in te sluiten. Het sturen van uw mannetje gebeurt met de joystick in poort twee. Een stukje muur verschijnt er links van uw mannetje als u op de vuurknop drukt. Denk erom het lijkt veel makkelijker dan het is.

```
10    rem ******************************
      **
20    rem *          lockin' man
       *
30    rem ******************************
      **
40    rem
50    rem         door luc de cock
60    rem        2990haacht (belgie)
70    rem
80    forx=49192to49236:reada:pokex,a:ne
      xt
90    data32,121,0,32,253,174,32,138,173
      ,32,191,177,166,100,164,101,96,32,
      40,192
100   data132,63,132,20,134,64,134,21,32
      ,19,166,165,95,56,233,1,133,65,165
      ,96,233
110   data0,133,66,96
120   poke198,0
130   rem
140   goto870
150   rem * hoofdlus spel **************
      *
160   ob=peek(53278):bo=0
170   poke53248,zx+x:poke53249,y+zy
180   poke53250,zx+s:poke53251,zy+p
190   poke53252,zx+s1:poke53253,zy+p1
200   if dc>5 thenpoke53254,zx+s2:poke53
      255,zy+p2
210   if dc=11 then 2750
220   ob=peek(53278):bo=0
230   if ti>=200then gosub 650
240   bo=peek(53278):ob=peek(53278):if b
      o<>0then 750
250   if peek(56320)=111 then gosub 590
260   rem * besturing *
270   a=peek(56320)
280   if a=119 then x=x+8:gosub 560:ifte
      =160then x=x-8
290   if a=123 then x=x-8:gosub 560:ifte
      =160then x=x+8
300   if a=125 then y=y+8:gosub 560:ifte
      =160then y=y-8
310   if a=126 then y=y-8:gosub 560:ifte
      =160then y=y+8
320   gosub560:if te=32 then 360
330   poke1024+(x/8)+40*int(y/8),32:ts=t
      s+10:pt=pt-1:print"[HOME][CTRL-2]"
      tab(22)"punten";ts
340   if pt=0thengr=10:sc=sc+1:dc=dc+1:
      ifdc=12thendc=1:sc=0:goto960
350   if pt=0then goto 960
360   on pp gosub 400,420,440,460
370   on pr gosub 480,500,520,540
380   if dc>5 then on pq gosub 2840,2860
      ,2880,2900
390   goto170
400   s=s-8:gosub570:ifte=160thens=s+8:
      pp=int(rnd(1)*2)+3:return
410   return
420   s=s+8:gosub570:ifte=160thens=s-8:
      pp=int(rnd(1)*2)+3:return
430   return
440   p=p-8:gosub570:ifte=160thenp=p+8:
      pp=int(rnd(1)*2)+1:return
450   return
460   p=p+8:gosub570:ifte=160thenp=p-8:
      pp=int(rnd(1)*2)+1:return
470   return
480   s1=s1-8:gosub 580:ifte=160thens1=
      s1+8:pr=int(rnd(1)*2)+3:return
490   return
500   s1=s1+8:gosub 580:ifte=160thens1=
      s1-8:pr=int(rnd(1)*2)+3:return
510   return
520   p1=p1-8:gosub 580:ifte=160thenp1=
      p1+8:pr=int(rnd(1)*2)+1:return
530   return
540   p1=p1+8:gosub 580:ifte=160thenp1=
      p1-8:pr=int(rnd(1)*2)+1:return
550   return
560   te=peek(1024+(x/8)+40*int(y/8)):re
      turn
570   te=peek(1024+(s/8)+40*int(p/8)):re
      turn
580   te=peek(1024+(s1/8)+40*int(p1/8)):
      return
590   rem * plaats speler blokje *******
      *
600   e=x-8:f=y:ts=ts-10
610   if peek(1024+(e/8)+40*int(f/8))=16
      0then return
620   if peek(1024+(e/8)+40*int(f/8))=46
       then pt=pt-1
630   poke 1024+(e/8)+40*int(f/8),160
640   return
650   rem * verplaats extra blokje *****
      *
660   poke1024+(q/8)+40*int(r/8),32
670   q=(int(rnd(1)*26)*8)+8
680   r=(int(rnd(1)*21)*8)+8
690   if peek(1024+(q/8)+40*int(r/8))=16
      0then 670
700   if peek(1024+(q/8)+40*int(r/8))=46
       then pt=pt-1
710   poke1024+(q/8)+40*int(r/8),160
720   eb=eb+10
730   ti$="000000"
740   return
750   rem * speleinde routine **********
      *
760   gosub 2920
770   if x+4>s and x-4<s and y+4>p and y
      -4<p then 800
780   if x+4>s1 and x-4<s1 and y+4>p1 an
      d y-4<p1 then 800
790   s1=120:p1=40:s2=16:p2=168:pq=1:pr=
```

```
     2:pp=2:forw=0to200:nextw:goto150
800  poke53269,0
810  print"[CTRL-2][SHIFT-CLR][3xCRSR-D
     OWN]"
820  print"[6xCRSR-RIGHT][COM-A][SHIFT-
     *][COM-S][COM-A][SHIFT-*][COM-S][C
     OM A][COM-R][COM-S][COM-A][SHIFT-*
     ][COM-S][3xSPACE][COM-A][SHIFT-*][
     2xCOM-S][SPACE][2xCOM-A][SHIFT-*][
     COM-S][COM-A][SHIFT-*][COM-S]"
830  print"[6xCRSR-RIGHT][SHIFT--][SPAC
     E][COM-S][COM-Q][SHIFT-*][COM-W][3
     xSHIFT--][COM-Q][SHIFT-*][4xSPACE]
     [SHIFT--][SPACE][2xSHIFT--][SPACE]
     [SHIFT--][COM-Q][SHIFT-*][SPACE][C
     OM Q][COM-R][COM-X]"
840  print"[6xCRSR-RIGHT][COM-Z][SHIFT-
     *][2xCOM-X][SPACE][COM-Z][COM-X][S
     PACE][2xCOM-Z][SHIFT-*][COM-X][3xS
     PACE][COM-Z][SHIFT-*][COM-X]J[SHIF
     T *]K[COM-Z][SHIFT-*][2xCOM-X][COM
      Z]"
850  print"[7xCRSR-DOWN][10xCRSR-RIGHT]
     u[SPACE]haalde";ts;"punten"
860  forw=0to1000:nextw:poke53269,0:run
     120
870  rem * intro-scherm ***************
     *
880  poke53280,0:poke53281,0:zx=24:zy=5
     0:dc=1:sc=0
890  print"[CTRL-2][SHIFT-CLR][22xCRSR-
     DOWN][8xCRSR-RIGHT]druk[SPACE]een[
     SPACE]toets[SPACE]voor[SPACE]start
     "
900  print"[3xCRSR-UP][6xCRSR-RIGHT]vuu
     rknop[SPACE]=[SPACE]stukje[SPACE]m
     uur[SPACE]links"
910  poke646,rnd(1)*15+1
920  print"[HOME][4xCRSR-DOWN][5xCRSR-R
     IGHT][COM-R][2xSPACE][COM-A][SHIFT
      *][COM-S][COM-A][SHIFT-*][2xCOM-S
     ][COM-A][COM-S][SPACE][COM-R][SPAC
     E][COM-A][SHIFT-*][2xCOM-S][2xSPAC
     E][COM-A][COM-R][COM-S][COM-A][SHI
     FT *][COM-S][COM-A][SHIFT-*][COM-S
     ]"
930  print"[5xCRSR-RIGHT][SHIFT--][2xSP
     ACE][SHIFT--][SPACE][2xSHIFT--][2x
     SPACE][COM-Q][COM-W][2xSPACE][SHIF
     T -][SPACE][SHIFT--][SPACE][SHIFT-
     -][3xSPACE][3xSHIFT--][COM-Q][SHIF
     T *][COM-W][SHIFT--][SPACE][SHIFT-
     -]"
940  print"[5xCRSR-RIGHT][COM-Z][SHIFT-
     *][COM-X][COM-Z][SHIFT-*][COM-X][C
     OM Z][SHIFT-*][2xCOM-X][COM-Z][COM
      X][SPACE][COM-E][SPACE][COM-X][SP
     ACE][COM-Z][3xSPACE][COM-X][SPACE]
     [COM-Z][COM-X][SPACE][COM-Z][COM-X
     ][SPACE][COM-Z]"
950  geta$:ob=peek(56320):ifa$="" and o
     b<>111 then 910
960  rem * schermopbouw ***************
     *
970  poke53269,0:print"[CTRL-6][SHIFT-C
     LR]"
980  pt=0:restore
990  rem
1000 if dc=11 thensys49209,2400:goto107
     0
1010 if sc=0thensys49209,980
1020 if sc=1 thensys49209,1710
1030 if sc=2 thensys49209,1940
1040 if sc=3 thensys49209,1400
1050 if sc=4 thensys49209,2150
1060 if sc=5 thensc=0:sys49209,980
1070 fora=1 to 22
1080 read a$:b$=""
1090 for b=1 to 27
1100 c$=mid$(a$,b,1)
1110 if c$="0" then c$="[CTRL-9][SPACE]
     [CTRL-0]"
1120 if c$="[SPACE]" then c$="[SPACE]"
1130 if c$="." then c$=".":pt=pt+1
1140 b$=b$+c$
1150 next b
1160 print"[SPACE]";b$
1170 next a
1180 print"[HOME][CTRL-2][SPACE]scherm"
     ;dc;tab(22)"punten";ts
1190 rem * sprites inlezen ************
     *
1200 sys49209,860
1210 forw=0to62:reada:poke12288+w,a:nex
     t:forw=0to62:reada:poke12352+w,a:n
     ext
1220 data60,0,0,126,0,0,219,0,0,255,0,0
     ,189,0,0,195,0,0,102,0,0,60,0,0,0,
     0,0
1230 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,0,0
1240 data66,0,0,60,0,0,90,0,0,90,0,0,12
     6,0,0,213,0,0,171,0,0,126,0,0,0,0,
     0
1250 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,0,0
1260 poke2040,192:poke2041,193:poke2042
     ,193:poke2043,193
1270 poke53287,7:poke53288,1:poke53289,
     3:poke53290,10
1280 rem
1290 rem * init ***********************
     *
1300 x=112:y=128:s=120:p=72:s1=104:p1=7
     2:pr=2:pp=1:s2=16:p2=168:pq=2:ti$=
     "000000
1310 ifdc=11thens=176:p=40:s1=48:p1=56:
     s2=176:p2=72
1320 ob=peek(53278):ob=peek(53278)
1330 q=10:r=10
1340 gosub 670
1350 poke53269,7:ifdc>5thenpoke53269,15
1360 goto 150
1370 rem * scherm opbouw **************
     *
1380 rem  123456789012345678901234567
1390 data"000000000000000000000000000"
1400 data"0...........000...........0"
1410 data"0.000000.00.....00.000000.0"
1420 data"0.0....0....000....0....0.0"
1430 data"0.0.00.0.000[3xSPACE]000.0.00
     .0.0"
1440 data"0...00......000......00...0"
1450 data"0000000.0.0.000.0.0.0000000"
1460 data"0.......0....0....0.......0"
1470 data"0.00000000.0.0.0.00000000.0"
1480 data"0..........0.0.0..........0"
1490 data"00.0.0000.00.0.00.0000.0.00"
1500 data"00.0....0.........0....0.00"
1510 data"00.0.00.0.0000000.0.00.0.00"
```

```
1520  data"0..0...................0..0"
1530  data"0.0000000.0000000.0000000.0"
1540  data"0.0...0.....0.......0...0.0"
1550  data"0...0.0.000.0.0.000.0.0...0"
1560  data"0.000.0.0.........0.0.000.0"
1570  data"0.........0.0.0.0.........0"
1580  data"0.000000000.0.0.000000000.0"
1590  data"0.............0...........0"
1600  data"000000000000000000000000000"
1610  rem  123456789012345678901234567
1620  data"000000000000000000000000000"
1630  data"0.............0...........0"
1640  data"0.0000.000000....00000000.0"
1650  data"0.0.....0....000........0.0"
1660  data"0.0.0.0...0.0[3xSPACE]0...0.0
      .0.0"
1670  data"0.0...000.0..0000.000...0.0"
1680  data"0.00000...00.000....00000.0"
1690  data"0.....0.0..0.0.0.0........0"
1700  data"0.000...00.0.0.0.00.0.0.000"
1710  data"0...0.0.0....0.0....0.0.0.0"
1720  data"0.0.0.0...0000...00.0.0.0.0"
1730  data"0.0.0.000.....0.....0.0.0.0"
1740  data"0.0.....000.00000.000.0...0"
1750  data"0.0.000.......0....0....000"
1760  data"0...0....0.00.0.00.0.0000.0"
1770  data"0.000.00.0.0.....0.0......0"
1780  data"0..0.....0.0.000...000.0.00"
1790  data"0.00.00000.......0.0......0"
1800  data"0......0...000.0.0...0.00.0"
1810  data"0.0.0....0...0...0.0.0.0..0"
1820  data"0...0..0...0.0.....0.....00"
1830  data"000000000000000000000000000"
1840  rem  123456789012345678901234567
1850  data"000000000000000000000000000"
1860  data"0............0....000.....0"
1870  data"0.0000000.00....0.....000.0"
1880  data"0..0....0.00000..00.00000.0"
1890  data"00.0.0.....0[3xSPACE]0.00.00
      ...0"
1900  data"00...0.00...000..00.00.0.00"
1910  data"00.0...0..0.000.000.0..0..0"
1920  data"0...000.00..0.......0.000.0"
1930  data"0.00.000.....0.0000.0.00..0"
1940  data"0.00.0...000.0.0000.0.00.00"
1950  data"0.00...00000.0.0000...00.00"
1960  data"0.00000..........00.0.00.00"
1970  data"0.00000.0.00.000.0.......00"
1980  data"0.....0.0........0.0.000000"
1990  data"000.0.0.0.00.0.0..........0"
2000  data"00......0.0.....00.0.000..0"
2010  data"0..000.0....000....0.000.00"
2020  data"0.0000.0.000...000.0......0"
2030  data"0......0...0.0.000.000000.0"
2040  data"0.00.0.000.0.0.....000000.0"
2050  data"0....0.......000..........0"
2060  data"000000000000000000000000000"
2070  rem  123456789012345678901234567
2080  data"000000000000000000000000000"
2090  data"0....0.....0.........0....0"
2100  data"0....0.........0.........00"
2110  data"0...........000..0....0...0"
2120  data"0...0......0[3xSPACE]0.......
      ...0"
2130  data"00.........0000...........0"
2140  data"0...0...0...00............0"
2150  data"0.........0..0............0"
2160  data"0.........0..0............0"
2170  data"0.........00000...........0"
2180  data"0.........0..............00"
2190  data"0......................0.00"
2200  data"0.........................0"
2210  data"0.......0...000...........0"
2220  data"0.........................0"
2230  data"00........................0"
2240  data"0..0......................0"
2250  data"0...................0.....0"
2260  data"0.........................0"
2270  data"0.........................0"
2280  data"0......0......0........0..0"
2290  data"000000000000000000000000000"
2300  rem  123456789012345678901234567
2310  data"000000000000000000000000000"
2320  data"0......000...000.........00"
2330  data"0.0000.0...0.....00.0000..0"
2340  data"0......0.0.000.0....0...0.0"
2350  data"0.000.00...0...000........0"
2360  data"0...0.0000000000...0000.0.0"
2370  data"0.0.0....0...0...00.....0.0"
2380  data"0.0...00.0.0.0.0000.000...0"
2390  data"0....0.0...0.0.0...0..0.000"
2400  data"000..0.0000..0.0.0.0..0...0"
2410  data"0.00.0....0.0000.0.0..0.0.0"
2420  data"0.........0......0....0...0"
2430  data"00...0..00000000.0000.0...0"
2440  data"00......0.........0...0...0"
2450  data"0.........0000000.0.00...00"
2460  data"0.0.......0.....0...00....0"
2470  data"0.0..0..000.000.0000......0"
2480  data"0............0....0.......0"
2490  data"000.00.00.0000000.0......00"
2500  data"0.........0.........0.....0"
2510  data"0.........0..00000....0...0"
2520  data"000000000000000000000000000"
2530  data"000000000000000000000000000"
2540  data"0[SPACE].....................
      ..[SPACE]0"
2550  data"0[SPACE]0...............0.0..
      .0[SPACE]0"
2560  data"0[SPACE]00000000000000000.000
      00[SPACE]0"
2570  data"0[SPACE]0[SPACE]0[17xSPACE]0.
      0[SPACE]0"
2580  data"0[SPACE]0[SPACE]00.0000000000
      000000.0[SPACE]0"
2590  data"0[SPACE]0[SPACE]0[17xSPACE]0.
      0[SPACE]0"
2600  data"0?0[SPACE]000000000000000.000
      .0?0"
2610  data"0[SPACE]0[SPACE]0............
      .....0.0[SPACE]0"
2620  data"0[SPACE]00000.000000000000000
      00[SPACE]0"
2630  data"0[3xSPACE]0[17xSPACE]0[3xSPAC
      E]0"
2640  data"0[SPACE]0000000000.0000000000
      00[SPACE]0"
2650  data"0[SPACE]0.....0........0.....
      .0[SPACE]0"
2660  data"0[SPACE]0.00000.........00000
      .0[SPACE]0"
2670  data"0[SPACE]0.0...0.0000000.0...0
      .0[SPACE]0"
2680  data"0[SPACE]0.0.0.0.........0.0.0
      .0[SPACE]0"
2690  data"0[SPACE]0.0.0.00000000000.0.0
      .0[SPACE]0"
2700  data"0[SPACE]0.0.0.00...0...00.0.0
      .0[SPACE]0"
2710  data"0[SPACE]0...0....0.0.0....0..
      .0[SPACE]0"
2720  data"0.0000000000.0.0000000000.0"
```

```
2730  data"0............0............0"
2740  data"000000000000000000000000000"
2750  if x<32andy>56andcc=0then2780
2760  if x>188andy>56andcc=1then2800
2770  goto 230
2780  poke1024+2+80,160:cc=1
2790  poke1024+9+40*13,32:goto2770
2800  poke1024+26+80,160
2810  poke1024+19+13*40,32
2820  poke1024+24+160,32:cc=2
2830  goto 2770
2840  s2=s2-8:gosub2950:ifte=160thens2=s
      2+8:pq=int(rnd(1)*2)+3:return
2850  return
2860  s2=s2+8:gosub2950:ifte=160thens2=s
      2-8:pq=int(rnd(1)*2)+3:return
2870  return
2880  p2=p2-8:gosub2950:ifte=160thenp2=p
      2+8:pq=int(rnd(1)*2)+1:return
2890  return
2900  p2=p2+8:gosub2950:ifte=160thenp2=p
      2-8:pq=int(rnd(1)*2)+1:return
2910  return
2920  if dc<=5 then return
2930  if x+4>s2 and x-4<s2 and y+4>p2 an
      d y-4<p2 then800
2940  return
2950  te=peek(1024+(s2/8)+40*int(p2/8)):
      return
```

** EINDE LISTING lock-in

REGEL 10	207	REGEL 620	14	REGEL 1230	71	REGEL 1840	21	REGEL 2450	187
REGEL 20	92	REGEL 630	49	REGEL 1240	144	REGEL 1850	185	REGEL 2460	175
REGEL 30	207	REGEL 640	142	REGEL 1250	71	REGEL 1860	177	REGEL 2470	189
REGEL 40	143	REGEL 650	162	REGEL 1260	75	REGEL 1870	195	REGEL 2480	169
REGEL 50	80	REGEL 660	23	REGEL 1270	215	REGEL 1880	193	REGEL 2490	195
REGEL 60	5	REGEL 670	113	REGEL 1280	143	REGEL 1890	45	REGEL 2500	169
REGEL 70	143	REGEL 680	109	REGEL 1290	221	REGEL 1900	191	REGEL 2510	179
REGEL 80	148	REGEL 690	26	REGEL 1300	224	REGEL 1910	189	REGEL 2520	215
REGEL 90	245	REGEL 700	38	REGEL 1310	170	REGEL 1920	185	REGEL 2530	215
REGEL 100	54	REGEL 710	73	REGEL 1320	248	REGEL 1930	191	REGEL 2540	73
REGEL 110	169	REGEL 720	203	REGEL 1330	3	REGEL 1940	195	REGEL 2550	81
REGEL 120	149	REGEL 730	215	REGEL 1340	42	REGEL 1950	195	REGEL 2560	117
REGEL 130	143	REGEL 740	142	REGEL 1350	14	REGEL 1960	187	REGEL 2570	21
REGEL 140	40	REGEL 750	70	REGEL 1360	31	REGEL 1970	191	REGEL 2580	67
REGEL 150	199	REGEL 760	90	REGEL 1370	205	REGEL 1980	183	REGEL 2590	21
REGEL 160	12	REGEL 770	193	REGEL 1380	21	REGEL 1990	183	REGEL 2600	193
REGEL 170	55	REGEL 780	133	REGEL 1390	215	REGEL 2000	183	REGEL 2610	35
REGEL 180	27	REGEL 790	81	REGEL 1400	171	REGEL 2010	189	REGEL 2620	117
REGEL 190	129	REGEL 800	252	REGEL 1410	197	REGEL 2020	189	REGEL 2630	135
REGEL 200	38	REGEL 810	168	REGEL 1420	179	REGEL 2030	189	REGEL 2640	117
REGEL 210	155	REGEL 820	173	REGEL 1430	55	REGEL 2040	193	REGEL 2650	81
REGEL 220	12	REGEL 830	47	REGEL 1440	179	REGEL 2050	173	REGEL 2660	97
REGEL 230	236	REGEL 840	198	REGEL 1450	203	REGEL 2060	215	REGEL 2670	99
REGEL 240	37	REGEL 850	165	REGEL 1460	171	REGEL 2070	21	REGEL 2680	89
REGEL 250	181	REGEL 860	191	REGEL 1470	203	REGEL 2080	215	REGEL 2690	107
REGEL 260	150	REGEL 870	212	REGEL 1480	171	REGEL 2090	171	REGEL 2700	95
REGEL 270	6	REGEL 880	131	REGEL 1490	199	REGEL 2100	171	REGEL 2710	87
REGEL 280	249	REGEL 890	68	REGEL 1500	177	REGEL 2110	175	REGEL 2720	207
REGEL 290	244	REGEL 900	165	REGEL 1510	199	REGEL 2120	33	REGEL 2730	167
REGEL 300	250	REGEL 910	141	REGEL 1520	169	REGEL 2130	175	REGEL 2740	215
REGEL 310	251	REGEL 920	137	REGEL 1530	207	REGEL 2140	173	REGEL 2750	174
REGEL 320	221	REGEL 930	70	REGEL 1540	175	REGEL 2150	169	REGEL 2760	226
REGEL 330	217	REGEL 940	108	REGEL 1550	189	REGEL 2160	169	REGEL 2770	30
REGEL 340	178	REGEL 950	46	REGEL 1560	185	REGEL 2170	175	REGEL 2780	178
REGEL 350	224	REGEL 960	247	REGEL 1570	173	REGEL 2180	169	REGEL 2790	131
REGEL 360	158	REGEL 970	196	REGEL 1580	205	REGEL 2190	169	REGEL 2800	69
REGEL 370	165	REGEL 980	76	REGEL 1590	167	REGEL 2200	165	REGEL 2810	33
REGEL 380	29	REGEL 990	143	REGEL 1600	215	REGEL 2210	173	REGEL 2820	228
REGEL 390	33	REGEL 1000	241	REGEL 1610	21	REGEL 2220	165	REGEL 2830	89
REGEL 400	120	REGEL 1010	30	REGEL 1620	215	REGEL 2230	167	REGEL 2840	117
REGEL 410	142	REGEL 1020	70	REGEL 1630	167	REGEL 2240	167	REGEL 2850	142
REGEL 420	120	REGEL 1030	76	REGEL 1640	201	REGEL 2250	167	REGEL 2860	117
REGEL 430	142	REGEL 1040	68	REGEL 1650	177	REGEL 2260	165	REGEL 2870	142
REGEL 440	106	REGEL 1050	72	REGEL 1660	45	REGEL 2270	165	REGEL 2880	103
REGEL 450	142	REGEL 1060	212	REGEL 1670	191	REGEL 2280	171	REGEL 2890	142
REGEL 460	106	REGEL 1070	173	REGEL 1680	195	REGEL 2290	215	REGEL 2900	103
REGEL 470	142	REGEL 1080	130	REGEL 1690	177	REGEL 2300	21	REGEL 2910	142
REGEL 480	63	REGEL 1090	179	REGEL 1700	193	REGEL 2310	215	REGEL 2920	225
REGEL 490	142	REGEL 1100	100	REGEL 1710	181	REGEL 2320	179	REGEL 2930	137
REGEL 500	63	REGEL 1110	192	REGEL 1720	189	REGEL 2330	189	REGEL 2940	142
REGEL 510	142	REGEL 1120	236	REGEL 1730	183	REGEL 2340	181	REGEL 2950	12
REGEL 520	49	REGEL 1130	87	REGEL 1740	191	REGEL 2350	183		
REGEL 530	142	REGEL 1140	143	REGEL 1750	181	REGEL 2360	197		
REGEL 540	49	REGEL 1150	196	REGEL 1760	189	REGEL 2370	179		
REGEL 550	142	REGEL 1160	126	REGEL 1770	183	REGEL 2380	191		
REGEL 560	35	REGEL 1170	195	REGEL 1780	187	REGEL 2390	183		
REGEL 570	21	REGEL 1180	228	REGEL 1790	183	REGEL 2400	189		
REGEL 580	119	REGEL 1190	26	REGEL 1800	183	REGEL 2410	189		
REGEL 590	80	REGEL 1200	112	REGEL 1810	181	REGEL 2420	171		
REGEL 600	3	REGEL 1210	194	REGEL 1820	177	REGEL 2430	195		
REGEL 610	243	REGEL 1220	252	REGEL 1830	215	REGEL 2440	173		

Pesten C 16

Dit maal stuurde Jos van Ooijen ons het spelletje pesten toe en omdat de spelregels nogal een verschillen heeft hij de spelregels in het menu opgenomen tijdens het lezen hoort men een leuk melodietje. Dit is weer zo'n verslavend spelletje dat op regenachtige zomeravonden menig uurtje zal doen verdwijnen. Programma pesten16

```
10   rem pesten c-16 & plus 4
20   rem door jos van oijen
40   rem boven-leeuwen
50   trap2450:fora=1to8:keya,"":next
60   color4,1:color0,1,5:scnclr:vol8:b=
     27::wc=13:ws=13:printchr$(27)+"m"
70   printchr$(142)chr$(8):poke1344,0
80   char,0,3,"[COM-6][COM-Q]DIUDIUDIC[
     COM-R]CUDI[COM-Q]DI[SPACE]UDI[SPAC
     E][COM-W]UDI[8xSPACE]B[SPACE]B"
90   char,0,4,"[CTRL-7][COM-Q]FK[COM-Q]
     CKJCI[SPACE]B[SPACE][COM-Q]CKB[SPA
     CE]B[SPACE]G[2xSPACE]CB[COM-Q]DI[S
     PACE]&[SPACE]plus[SPACE][COM-Z]C[S
     HIFT +]"
100  char,0,5,"[COM-7]B[2xSPACE]JFKJFK[
     SPACE]B[SPACE]JFKB[SPACE]B[SPACE]J
     FK[SPACE][COM-E]JFK[10xSPACE]B"
110  char,11,8,"[CTRL-2]door[SPACE]jos[
     SPACE]van[SPACE]oijen":char,34,9,"
     W"
120  char,0,10,"[COM-4]UDIUDI[COM-Q]DI[
     2xSPACE][COM-Q]D[COM-R]DIUDI[COM-Q
     ]D[COM-R]DIUDI[COM-Q]DIC[COM-R]C[S
     PACE]BUDI"
130  char,0,11,"[CTRL-5][COM-Q]CK[COM-Q
     ]CKB[SPACE]B[2xSPACE]B[SPACE]B[SPA
     CE]BG[SPACE]HB[SPACE]B[SPACE]B[COM
      Q]CKB[SPACE]B[SPACE]B[2xSPACE]B[C
     OM Q]CK"
140  char,0,12,"[CTRL-3]JFKJFKB[SPACE]B
     [2xSPACE]B[SPACE]B[SPACE]BJFKB[SPA
     CE]B[SPACE]BJFKB[SPACE]B[SPACE]BJF
     KJFK"
150  char,4,14,"[COM-5]UDIUDI[COM-A]DIB
     [SPACE]BB[SPACE][COM-A]DI[2xSPACE]
     UDI[SPACE]B[SPACE]B[SPACE][COM-A]D
     I"
160  char,4,15,"[COM-1]G[SPACE][COM-S][
     COM-Q]CKB[SPACE]HB[SPACE]BB[SPACE]
     B[SPACE]H[2xSPACE][COM-Q]C[COM-W][
     SPACE]B[SPACE]B[SPACE][COM-Q]C[COM
      W]"
170  char,4,16,"[COM-2]JFKJFK[COM-Z]FKJ
     FK[COM-Z]C[COM-Z]FK[2xSPACE]B[SPAC
     E]BWJFKW[COM-Z]FKW"
180  char,16,23,"[CTRL-8](c)[SPACE]1987
     "
190  dimc$(108),b$(108),s$(32)
200  d=1:fora=1to4:readz$(a):nexta
210  forl=1to52step13:fora=1tol+13:c$(a
     +d-1)=z$(l/13+1)+str$(a-l+1):nexta
     ,l
220  data"S","Z","A","X"
230  ifd<>55thend=55:goto210
240  c$(53)="j1":c$(54)="j2":c$(107)="j
     3":c$(108)="j4"
250  char,12,20,"(effe[SPACE]schudde..)
     "
260  rem  kaarten schudden en delen
270  a=rnd(-ti):fora=1to108
280  l=int(rnd(1)*108+1):b$(a)=c$(l):c$
     (l)=""
290  ifb$(a)=""then280
300  forr=1to7:poke65287,r:nextr
310  nexta
320  fora=1to13:c$(a)=b$(a):next
330  fora=14to26:s$(a-13)=b$(a):next
340  char,8,20,"[CTRL-5]bedankt[SPACE]v
     oor[SPACE]het[SPACE]wachten.[CTRL-
     1]":fora=1to200step4:sound1,a+800,
     1:next
350  fora=1to100step2:sound1,1000-a,1:n
     ext:poke65287,8:a=0
360  a=a+1:ifa=8thena=0
370  color1,8,a:char,7,20,"wil[SPACE]je
     [SPACE]instructies[SPACE](j/n)[SPA
     CE]?[CTRL-1]"
380  getg$:ifg$<>"j"andg$<>"n"then360
390  sound1,800,5:color0,2,4:ifg$="j"th
     en1950
400  rem  kaarten speler printen
410  scnclr
420  fora=0to39:char,a,8,"C":next
430  forp=0to39step5:n=p/5+1:k$=s$(n):z
     =9:gosub510:char,p+2,9,"[CTRL-2]"+
     chr$(48+n)+"[CTRL-1]":next
440  forp=0to24step5:n=p/5+9:k$=s$(n):z
     =13:gosub510
450  ifn>9thenn=n-9:char,p+2,13,"[CTRL-
     2]1"+chr$(47+n)+"[CTRL-1]":elsecha
     r,p+2,13,"[CTRL-2]"+chr$(48+n)+"[C
     TRL 1]"
460  nextp:gosub1710
470  rem
480  k$=b$(27):b$=k$:p=17:z=1:gosub510
490  ifkl$="j"orkr$="a"orkr$="2"then139
     0:else700
500  rem  kaart printen
510  l=p+3072+z*40:pokel,85:pokel+1,67:
     pokel+3,67:pokel+4,73:ifp=17thenpo
     kel+2,67
520  ifpeek(l+160)=85orl>3911thenm=3:el
     sem=5
530  fora=l+40tol+m*40step40:pokea,66:p
     okea+4,66:nexta:ifm=3then550
540  pokel+240,74:pokel+241,67:pokel+24
     2,67:pokel+243,67:pokel+244,75
550  kl$=left$(k$,1):ifkl$="S"orkl$="Z"
     thenkt=3:elsekt=1
560  forl=1tom:char,p+1,z+l,"[3xSPACE]"
     :next
570  ifkl$="j"thenchar,p+2,z+1,"j[CRSR
     -DOWN][CRSR-LEFT]o[CRSR-DOWN][CRSR
     -LEFT]k[CRSR-DOWN][CRSR-LEFT]e[CRS
     R-DOWN][CRSR-LEFT]r":goto680
580  kr$=right$(k$,2)
590  ifkr$="10"then650
600  ifkr$="11"thenkr$="b":goto650
610  ifkr$="12"thenkr$="v":goto650
620  ifkr$="13"thenkr$="h":goto650
630  kr$=right$(kr$,1)
640  ifkr$="1"thenkr$="a"
650  char,p+1,z+1,kr$:color1,kt,3:char,
     p+3,z+1,kl$
660  ifm=3then680
670  char,p+1,z+5,kl$:color1,1:ifkr$<>"
     10"thenchar,p+2,z+5,"[SPACE]"+kr$:
     elsechar,p+2,z+5,kr$
680  return
690  rem  speler aan de beurt
```

```
700  sound1,300,5:char,26,4,"[CTRL-9]jouw[SPACE]beurt[CTRL-0]":char,16,2,chr$(27)+"p":char,16,3,chr$(27)+"p"
710  print"[HOME][9xCRSR-RIGHT]"chr$(27)"b"
720  printchr$(27)"j[6xCRSR-RIGHT]"chr$(27)"q"
730  input"kaart";k:ifk<0ork>32then720
740  rem  kopen
750  ifk<>0then800
760  kk=kk+1:ifkk=2thenchar,0,2,"je[SPACE]kunt[SPACE]dus[SPACE]niet.":kk=0:fora=1to800:next:goto1230
770  b=b+1:ifb>108then2410
780  gosub1810
790  char,0,2,"je[SPACE]hebt[SPACE]gekocht.":fora=1to800:next:ws=ws+1:goto700
800  ss$=s$(k)
810  ifleft$(ss$,1)<>"j"then870
820  goto880
830  char,0,2,"o[SPACE]jee![SPACE]ik[SPACE]moet[SPACE]5":char,0,3,"kaarten[SPACE]pakken."
840  fora=600to0step-20:sound1,a,1:next
850  forc=1to5:b=b+1:wc=wc+1:gosub1770:next:fora=1to400:next:char,16,2,chr$(27)+"p"
860  char,16,3,chr$(27)+"p":gosub1530:gosub1710:goto700
870  ifleft$(ss$,1)<>left$(b$,1)andright$(s$(k),2)<>right$(b$,2)then720
880  ws=ws-1:k$=ss$:p=17:z=1:gosub510:s$(k)="":b$=ss$:kk=0
890  ifk<9thenp=k*5-4:z=8
900  ifk>8andk<17thenp=(k-8)*5-4:z=12
910  ifk>16andk<25thenp=(k-16)*5-4:z=16
920  ifk>24thenp=(k-24)*5-4:z=20
930  pp=3071+p+(z+5)*40
940  ifpeek(pp)=85orpp>3951thenl=4:elsel=7
950  fora=1tol:char,p-1,z+a,"[4xSPACE]":g=40*(z+a)+p+3075:pokeg,32:nexta
960  ifk-8<1then1040
970  ifs$(k-8)=""then1040
980  ifk<17thenz=9:p=(k-9)*5
990  ifk>16andk<25thenz=13:p=(k-17)*5
1000 ifk>24thenz=17:p=(k-25)*5
1010 g$=str$(k-8)
1020 g$="[CTRL-2]"+right$(g$,len(g$)-1)+"[CTRL-1]"
1030 k$=s$(k-8):gosub510:char,p+2,z,g$
1040 ifws=0then1930
1050 ifleft$(ss$,1)="j"then830
1060 ifright$(ss$,2)="11"then1140
1070 ifright$(ss$,2)<>"[SPACE]2"then1110
1080 char,0,2,"ik[SPACE]moet[SPACE]2[SPACE]kaarten":char,0,3,"erbij[SPACE]pakken.":fora=600to0step-20
1090 sound1,a,1:next
1100 forc=1to2:b=b+1:wc=wc+1:gosub1770:next:gosub1710:fora=1to800:next:goto700
1110 ifright$(ss$,1)="1"thenchar,0,2,"ik[SPACE]moet[SPACE]een[SPACE]beurt":char,0,3,"overslaan.":else1230
1120 fora=400to0step-20:sound1,a,1:next
1130 fora=1to800:next:goto700
1140 char,0,2,"wat[SPACE]maak[SPACE]je[SPACE]ervan"
1150 fora=1to2:char,1,a+2,str$(a)+"[SPACE][CTRL-3]"+z$(a)+"[CTRL-1]":next
1160 fora=3to4:char,1,a+2,str$(a)+"[SPACE]"+z$(a):next
1170 getkeya$:kl=val(a$):ifkl<1orkl>4then1170
1180 ifkl<3thencolor1,3,3
1190 sound1,800,5:char,12,4,z$(kl):color1,1:fora=1to800:next
1200 fora=2to6:char,16,a,chr$(27)+"p":next
1210 b$=z$(kl)+"[SPACE]"+right$(ss$,2)
1220 rem  computer aan de beurt
1230 kk=0:fora=0to3:char,16,a,chr$(27)+"p":next
1240 sound1,900,5:char,26,4,"mijn[SPACE]beurt"
1250 fora=1to32:ifleft$(c$(a),1)="j"thene=a
1260 next
1270 ifws<5ande<>0thenk=e:e=0:goto1380
1280 k=0:fora=1to32:ifleft$(c$(a),1)=left$(b$,1)thenk=a
1290 next:ifk<>0then1380
1300 k=0:fora=1to32:ifright$(c$(a),2)=right$(b$,2)thenk=a
1310 next:ifk<>0then1380
1320 ifwc<5ande<>0thenk=e:e=0:goto1380
1330 char,0,2,"ik[SPACE]koop.":fora=1to800:next:b=b+1:gosub1770
1340 wc=wc+1:gosub1710
1350 ifleft$(b$(b),1)=left$(b$,1)orright$(b$(b),2)=right$(b$,2)thenk=a:goto1380
1360 char,0,2,"ik[SPACE]kan[SPACE]niet.":fora=1to800:next:goto700
1370 rem  weggooien kaart
1380 wc=wc-1:k$=c$(k):b$=k$:p=17:z=1:gosub510:c$(k)="":gosub1710
1390 ifleft$(k$,1)<>"j"then1440
1400 char,0,2,"ha[SPACE]ha,[SPACE]je[SPACE]moet[SPACE]5":char,0,3,"kaarten[SPACE]pakken!":fora=600to1000step20
1410 sound1,a,1:next:forqq=1to5:b=b+1:ws=ws+1:gosub1810:nextqq
1420 fora=1to800:next:char,16,2,chr$(27)+"p"
1430 char,16,3,chr$(27)+"p":goto1140
1440 ifright$(k$,2)="11"thengosub1530:goto700
1450 ifright$(k$,2)<>"[SPACE]2"then1490
1460 char,0,2,"je[SPACE]moet[SPACE]2[SPACE]kaarten":char,0,3,"erbij[SPACE]pakken!"
1470 fora=600to1000step20:sound1,a,1:next
1480 forqq=1to2:s=s+1:b=b+1:ws=ws+1:gosub1810:nextqq:fora=1to800:next:goto1230
1490 ifright$(k$,1)="1"thenchar,0,2,"je[SPACE]moet[SPACE]een[SPACE]beurt":char,0,3,"overslaan!":else700
1500 fora=600to1000step20:sound1,a,1:next
1510 fora=1to800:next:goto1230
1520 rem  keuze kaart soort
1530 cc$=b$:ha=0:ru=0:kl=0:sc=0:fora=1to32:l$=left$(c$(a),1)
1540 ifl$="S"thenha=ha+1
```

```
1550  ifl$="Z"thenru=ru+1
1560  ifl$="X"thenkl=kl+1
1570  ifl$="A"thensc=sc+1
1580  next:q=ha
1590  ifq<ruthenq=ru
1600  ifq<klthenq=kl
1610  ifq<scthenq=sc
1620  ifq=hathenb$="[CTRL-3]S[CTRL-1]"
1630  ifq=ruthenb$="[CTRL-3]Z[CTRL-1]"
1640  ifq=klthenb$="[CTRL-1]X"
1650  ifq=scthenb$="[CTRL-1]A"
1660  char,0,2,"ik[SPACE]maak[SPACE]er[S
      PACE]"+b$
1670  char,13,2,"van.":fora=1to1000:next
1680  b$=mid$(b$,2,1)+"[SPACE]"+right$(c
      c$,2)
1690  return
1700  rem
1710  ifb>108then2410
1720  t$="[SPACE]kaarten[SPACE]":ifwc=1t
      hent$="[SPACE]kaart[2xSPACE]"
1730  char,0,7,"comp."+str$(wc)+t$
1740  ifwc=0thenchar,12,8,"[CTRL-2]ik[SP
      ACE]heb[SPACE]gewonnen[CTRL-1]":go
      to2300
1750  return
1760  rem lege plaatsen opvullen comp
1770  a=0:ifb>108then2410
1780  a=a+1:ifa<33andc$(a)<>""then1780
1790  c$(a)=b$(b):return
1800  rem  lege plaatsen opvullen speler
1810  a=0:ifb>108then2410
1820  a=a+1:ifa<33ands$(a)<>""then1820
1830  s$(a)=b$(b):s=a
1840  ifs<9thenp=(s-1)*5:z=9
1850  ifs>8ands<17thenp=(s-9)*5:z=13
1860  ifs>16thenp=(s-17)*5:z=17
1870  ifs>24thenp=(s-25)*5:z=21
1880  g$=str$(s)
1890  g$="[CTRL-2]"+right$(g$,len(g$)-1)
      +"[CTRL-1]"
1900  k$=s$(s):gosub510:char,p+2,z,g$
1910  return
1920  rem
1930  char,6,16,"[CTRL-2]proficiat,[SPAC
      E]je[SPACE]hebt[SPACE]gewonnen![CT
      RL 1]":goto2300
1940  rem   instructies
1950  scnclr:printchr$(14)
1960  char,13,0,"[CTRL-2][COM-A][12xSHIF
      T *][COM-S]"
1970  char,13,1,"[SHIFT--][CTRL-1]INSTRU
      CTIES:[CTRL-2][SHIFT--]"
1980  char,13,2,"[COM-Z][12xSHIFT-*][COM
       X][CTRL-1]"
1990  char,1,0,chr$(27)+"t":print
2000  print"[4xCRSR-DOWN]Het[SPACE]is[SP
      ACE]de[SPACE]bedoeling[SPACE]om[SP
      ACE]al[SPACE]je[SPACE]kaarten[3xSP
      ACE]kwijt[SPACE]te[SPACE]raken.":p
      rint
2010  print"Dit[SPACE]kan[SPACE]op[SPACE
      ]de[SPACE]volgende[SPACE]wijze:":p
      rint
2020  print"Als[SPACE]er[SPACE]b.v.[SPAC
      E]een[SPACE]harten[SPACE]3[SPACE]l
      igt,[SPACE]kun":print"je[SPACE]daa
      r[SPACE]een[SPACE]andere[SPACE]har
      ten[SPACE]";
2030  print"opgooien[SPACE]of[2xSPACE]ee
      n[SPACE]andere[SPACE]drie[SPACE]do
      or[SPACE]het[SPACE]nummer[SPACE]va
      n"
2040  print"die[SPACE]kaart[SPACE]in[SPA
      CE]te[SPACE]tikken[SPACE]en[SPACE]
      dan[SPACE]op":print"RETURN[SPACE]t
      e[SPACE]drukken.":print
2050  print"Als[SPACE]je[SPACE]niet[SPAC
      E]kunt,[SPACE]kun[SPACE]je[SPACE]e
      en[SPACE]kaart[5xSPACE]kopen[SPACE
      ]door[SPACE]een[SPACE]0[SPACE]in[S
      PACE]te[SPACE]tikken.":print
2060  print"Wanneer[SPACE]je[SPACE]dan[S
      PACE]nog[SPACE]niet[SPACE]kunt,[SP
      ACE]tik[SPACE]je[3xSPACE]nog[SPACE
      ]een[SPACE]0[SPACE]in;[SPACE]de[SP
      ACE]computer[SPACE]is[SPACE]dan."
2070  print:char,12,22,"DRUK[SPACE]OP[SH
      IFT SPACE]EEN[SPACE]TOETS"
2080  restore2460:do:readx,y:f=(111840.4
      5/(1024-x))/2:aa=1024-(111840.45/f
      )
2090  sound1,x,y:sound2,aa,y-3:forv=8to5
      step-1:fora=5toy:nexta:volv:nextv
2100  getg$:ifg$<>""then2120
2110  loopuntilx=0:goto2080
2120  scnclr:char,1,0,chr$(27)+"t"
2130  print"Als[SPACE]je[SPACE]een[SPACE
      ]aas[SPACE]opgooit,[SPACE]dan[SPAC
      E]moet[SPACE]de[4xSPACE]computer[S
      PACE]een[SPACE]beurt[SPACE]oversla
      an[SPACE]en[SPACE]";
2140  print"dan[4xSPACE]ben[SPACE]jij[SP
      ACE]weer.":print:print
2150  print"Als[SPACE]je[SPACE]een[SPACE
      ]boer[SPACE]opgooit[SPACE]kun[SPAC
      E]je[SPACE]ervan[3xSPACE]maken[SPA
      CE]wat[SPACE]je[SPACE]wilt,[SPACE]
      harten,[SPACE]ruiten,"
2160  print"klaveren[SPACE]of[SPACE]scho
      ppen."
2170  print"Dit[SPACE]doe[SPACE]je[SPACE
      ]door[SPACE]het[SPACE]cijfer[SPACE
      ]voor[SPACE]de[5xSPACE]soort[SPACE
      ]van[SPACE]je[SPACE]keuze[SPACE]in
      [SPACE]te[SPACE]tikken."
2180  print"Om[SPACE]een[SPACE]boer[SPAC
      E]op[SPACE]te[SPACE]kunnen[SPACE]g
      ooien,[SPACE]moet[2xSPACE]deze[SPA
      CE]wel[SPACE]van[SPACE]de[SPACE]ze
      lfde[SPACE]soort[SPACE]zijn[SPACE]
      ";
2190  print"als[2xSPACE]de[SPACE]kaart[S
      PACE]die[SPACE]er[SPACE]al[SPACE]l
      ag.":print:print
2200  print"Wanneer[SPACE]je[SPACE]een[S
      PACE]joker[SPACE]opgooit,[SPACE]mo
      et[SPACE]de[2xSPACE]computer[SPACE
      ]vijf[SPACE]kaarten[SPACE]pakken,[
      SPACE]er[SPACE]van"
2210  print"maken[SPACE]wat[SPACE]hij[SP
      ACE]wil[SPACE]en[SPACE]dan[SPACE]m
      ag[SPACE]jij[SPACE]weer.":print"Ee
      n[SPACE]joker[SPACE]kun[SPACE]je[S
      PACE]altijd[SPACE]opgooien."
2220  print"[2xCRSR-DOWN]Wanneer[SPACE]j
      e[SPACE]een[SPACE]twee[SPACE]opgoo
      it,[SPACE]moet[SPACE]de[3xSPACE]co
      mputer[SPACE]twee[SPACE]kaarten[SP
      ACE]pakken,[SPACE]";
2230  print"daarna[3xSPACE]ben[SPACE]je[
      SPACE]zelf[SPACE]weer[SPACE]aan[SP
```

```
ACE]de[SPACE]beurt."
2240 do:readx,y:f=(111840.45/(1024-x))/
     2:aa=1024-(111840.45/f)
2250 sound1,x,y:sound2,aa,y-3:forv=8to5
     step-1:fora=5toy:nexta:volv:nextv
2260 getg$:ifg$=chr$(32)then2280
2270 loopuntilx=0:restore2460:goto2240
2280 printchr$(27)+"n"chr$(142):vol8:go
     to410
2290 rem    eind
2300 a=0
2310 a=a+20:ifa>520then2330
2320 forb=700to1020stepa:sound1,b,2:nex
     t:goto2310
2330 a=0
2340 a=a+1:ifa=8thena=0
2350 color1,8,a:char,10,18,"nog[SPACE]e
     en[SPACE]keer[SPACE](j/n)[SPACE]?"
2360 getg$:ifg$<>"j"andg$<>"n"then2340
2370 sound1,800,5:ifg$="j"thenrun:elsec
     har,7,14,"[CTRL-8]weet[SPACE]je[SP
     ACE]het[SPACE]zeker[SPACE](j/n)[SP
     ACE]?"
2380 getkeyg$:ifg$<>"j"andg$<>"n"then23
     80
2390 sound1,800,5:ifg$="j"thensys62116
2400 char,0,14,chr$(27)+"q":goto2360
2410 fora=0to7:color0,2,a:forb=1to20:ne
     xtb,a:fora=7to0step-1:color0,2,a
2420 forb=1to20:nextb,a:scnclr:char,7,1
     2,"[CTRL-2]sorry,[SPACE]de[SPACE]k
     aarten[SPACE]zijn[SPACE]op."
2430 char,10,14,"we[SPACE]beginnen[SPAC
     E]opnieuw."
2440 fora=0to3000:next:run
2450 resumenext
2460 data810,30,834,7.5,864,7.5,854,7.5
     ,834,7.5,881,15,881,15,881,7.5,897
     ,7.5,854,7.5
2470 data864,7.5,834,15,834,15,834,7.5,
     864,7.5,854,7.5,834,7.5,810,7.5,91
     7,7.5,911,7.5
2480 data897,7.5,881,7.5,864,7.5,854,7.
     5,834,7.5,810,30,834,7.5,864,7.5,8
     54,7.5,834,7.5
2490 data881,15,881,15,881,7.5,897,7.5,
     854,7.5,864,7.5,834,15,834,15,864,
     7.5,854,7.5
2500 data834,7.5,810,7.5,881,7.5,834,7.
     5,864,7.5,810,15,897,7.5,0,0
```

** EINDE LISTING pesten16

REGEL 10	211	REGEL 520	225	REGEL 1020	154	REGEL 1520	29	REGEL 2020	33
REGEL 20	9	REGEL 530	57	REGEL 1030	3	REGEL 1530	66	REGEL 2030	75
REGEL 40	75	REGEL 540	155	REGEL 1040	139	REGEL 1540	10	REGEL 2040	194
REGEL 50	253	REGEL 550	146	REGEL 1050	77	REGEL 1550	77	REGEL 2050	226
REGEL 60	58	REGEL 560	186	REGEL 1060	146	REGEL 1560	43	REGEL 2060	27
REGEL 70	145	REGEL 570	208	REGEL 1070	17	REGEL 1570	18	REGEL 2070	30
REGEL 80	109	REGEL 580	90	REGEL 1080	114	REGEL 1580	72	REGEL 2080	245
REGEL 90	174	REGEL 590	229	REGEL 1090	145	REGEL 1590	135	REGEL 2090	30
REGEL 100	233	REGEL 600	162	REGEL 1100	214	REGEL 1600	103	REGEL 2100	80
REGEL 110	26	REGEL 610	183	REGEL 1110	186	REGEL 1610	101	REGEL 2110	175
REGEL 120	128	REGEL 620	170	REGEL 1120	93	REGEL 1620	153	REGEL 2120	170
REGEL 130	97	REGEL 630	171	REGEL 1130	247	REGEL 1630	190	REGEL 2130	153
REGEL 140	207	REGEL 640	18	REGEL 1140	27	REGEL 1640	144	REGEL 2140	105
REGEL 150	128	REGEL 650	188	REGEL 1150	187	REGEL 1650	120	REGEL 2150	42
REGEL 160	145	REGEL 660	2	REGEL 1160	37	REGEL 1660	95	REGEL 2160	88
REGEL 170	119	REGEL 670	121	REGEL 1170	217	REGEL 1670	81	REGEL 2170	239
REGEL 180	127	REGEL 680	142	REGEL 1180	133	REGEL 1680	14	REGEL 2180	103
REGEL 190	172	REGEL 690	53	REGEL 1190	81	REGEL 1690	142	REGEL 2190	87
REGEL 200	172	REGEL 700	1	REGEL 1200	65	REGEL 1700	143	REGEL 2200	96
REGEL 210	189	REGEL 710	252	REGEL 1210	88	REGEL 1710	133	REGEL 2210	34
REGEL 220	93	REGEL 720	108	REGEL 1220	217	REGEL 1720	62	REGEL 2220	1
REGEL 230	250	REGEL 730	6	REGEL 1230	238	REGEL 1730	231	REGEL 2230	85
REGEL 240	90	REGEL 740	12	REGEL 1240	95	REGEL 1740	210	REGEL 2240	99
REGEL 250	48	REGEL 750	169	REGEL 1250	1	REGEL 1750	142	REGEL 2250	30
REGEL 260	222	REGEL 760	214	REGEL 1260	130	REGEL 1760	168	REGEL 2260	222
REGEL 270	99	REGEL 770	208	REGEL 1270	126	REGEL 1770	226	REGEL 2270	63
REGEL 280	125	REGEL 780	87	REGEL 1280	188	REGEL 1780	245	REGEL 2280	170
REGEL 290	186	REGEL 790	59	REGEL 1290	153	REGEL 1790	108	REGEL 2290	175
REGEL 300	250	REGEL 800	143	REGEL 1300	192	REGEL 1800	68	REGEL 2300	35
REGEL 310	195	REGEL 810	3	REGEL 1310	153	REGEL 1810	226	REGEL 2310	253
REGEL 320	70	REGEL 820	41	REGEL 1320	110	REGEL 1820	0	REGEL 2320	179
REGEL 330	157	REGEL 830	127	REGEL 1330	189	REGEL 1830	52	REGEL 2330	35
REGEL 340	247	REGEL 840	95	REGEL 1340	81	REGEL 1840	83	REGEL 2340	201
REGEL 350	138	REGEL 850	225	REGEL 1350	173	REGEL 1850	160	REGEL 2350	207
REGEL 360	201	REGEL 860	55	REGEL 1360	7	REGEL 1860	229	REGEL 2360	174
REGEL 370	35	REGEL 870	199	REGEL 1370	166	REGEL 1870	222	REGEL 2370	63
REGEL 380	126	REGEL 880	144	REGEL 1380	28	REGEL 1880	133	REGEL 2380	171
REGEL 390	37	REGEL 890	244	REGEL 1390	210	REGEL 1890	154	REGEL 2390	229
REGEL 400	128	REGEL 900	101	REGEL 1400	27	REGEL 1900	40	REGEL 2400	71
REGEL 410	232	REGEL 910	198	REGEL 1410	255	REGEL 1910	142	REGEL 2410	70
REGEL 420	142	REGEL 920	171	REGEL 1420	147	REGEL 1920	143	REGEL 2420	28
REGEL 430	225	REGEL 930	91	REGEL 1430	70	REGEL 1930	99	REGEL 2430	165
REGEL 440	98	REGEL 940	77	REGEL 1440	33	REGEL 1940	236	REGEL 2440	139
REGEL 450	215	REGEL 950	218	REGEL 1450	193	REGEL 1950	56	REGEL 2450	88
REGEL 460	98	REGEL 960	9	REGEL 1460	146	REGEL 1960	159	REGEL 2460	114
REGEL 470	143	REGEL 970	227	REGEL 1470	69	REGEL 1970	163	REGEL 2470	153
REGEL 480	122	REGEL 980	122	REGEL 1480	254	REGEL 1980	56	REGEL 2480	217
REGEL 490	225	REGEL 990	229	REGEL 1490	30	REGEL 1990	91	REGEL 2490	74
REGEL 500	34	REGEL 1000	211	REGEL 1500	69	REGEL 2000	130	REGEL 2500	210
REGEL 510	113	REGEL 1010	96	REGEL 1510	38	REGEL 2010	42		

De listings voor de Amiga programma's moeten worden ingetypt na het opstarten van de Amigabasic. Deze wordt meegeleverd met uw computer op de "extra" schijf. Let erop hele regels invoeren en alleen aan het eind van elke regel een return geven. Dit hebben we aangegeven met [return]. De lege regels tussen de diverse blokken staan er alleen voor de duidelijkheid en hoeven niet te worden overgenomen (dit mag wel).
[return] *mag dus niet worden overgetypt maar moet uitgevoerd worden door op de returntoets te drukken.*

Bioritme

Na het opstarten van het programma wordt u gevraagd om uw naam, geboortedatum, datum voor de berekenig in te voeren. Deze data moet worden ingevoerd gescheiden door een komma. Bijvoorbeeld 29 maart 1988 zou worden 29,4,88. Hierna wordt uw fysieke, gevoels en intellectuele curve berekend. Hierna kunt u kiezen of u ook de volgende maand of mischien wel een maand terug wilt gaan bekijken. Met dit programma kunt u uitrekenen wat de beste dag is om bepaalde zaken te gaan uitvoeren. M.J. Kemps is de maker van dit programma.

```
REM >> Bioritme Amiga << [return]
REM - Door M.J. Kemps uit Schiedam -
[return]

CLEAR:IF FRE(x)<25000 THEN CLEAR ,
125000& [return]

initialisatie: [return]
DIM f(32),g(32),v(32),a(32),tm(12),m$
(12) [return]
pi=3.14159265# [return]
FOR i=1 TO 12:READ tm(i):NEXT [return]
DATA 31,28,31,30,31,30,31,31,30,31,30,
31 [return]
FOR r=1 TO 12:READ m$(r):NEXT [return]
DATA januari,februari,maart,april,mei,
juni [return]
DATA juli,augustus,september,oktober,
november,december [return]

startscherm: [return]
PALETTE 0,0,0,0 [return]
PALETTE 1,1,1,0 [return]
CLS:LOCATE 2,30:PRINT ">>>>> BIORITME
<<<<<" [return]
LOCATE 5,5:LINE INPUT "Wat is Uw
naam : ";n$ [return]
LOCATE 10,5:INPUT "Geboorte datum (d,
m,j)";gd,gm,gj [return]
IF gj<100 THEN gj=gj+1900 [return]
LOCATE 15,5:INPUT "Bioritme voor (m,j)
";m,j [return]
IF j<1800 THEN j=j+1900 [return]
IF m>12 OR gm>12 OR m<1 OR gm<1
THEN CLS:PRINT "foutieve datum-invoer":
GOTO alarm [return]
IF j>gj THEN berekening [return]
IF m>=gm THEN berekening [return]
CLS:LOCATE 2,1:PRINT "Toen was U nog
niet geboren !" [return]

alarm: [return]
BEEP:FOR w=1 TO 5000:NEXT w [return]
GOTO startscherm [return]

berekening: [return]
tot=0 [return]
IF gj=j AND gm=m THEN tot=0-(gd-1):GOTO
berekening2 [return]
FOR telj=gj TO j [return]
IF telj=gj OR telj=j THEN sub1 [return]
tot=tot+365 [return]
IF telj/4=INT(telj/4) THEN tot=tot+1
[return]
GOTO sub3 [return]

sub1: [return]
FOR telm=1 TO 12 [return]
ma=tm(telm):IF telm=2 AND telj/4=INT
(telj/4) THEN ma=ma+1 [return]
IF telj=gj AND telm<gm THEN sub4 [return]
IF telj=j AND telm>=m THEN sub3  [return]
IF telj=gj AND telm=gm THEN sub2 [return]
tot=tot+ma:GOTO sub4 [return]

sub2: [return]
tot=tot+(ma-(gd-1)) [return]

sub4: [return]
NEXT telm [return]

sub3: [return]
NEXT telj [return]

berekening2: [return]
mm=tm(m):IF m=2 AND j/4=INT(j/4) THEN
mm=mm+1 [return]
FOR i=tot TO tot+mm [return]
ritme=i-(tot-1) [return]
f(ritme)=SIN(2*pi/23*i) [return]
g(ritme)=SIN(2*pi/28*i) [return]
v(ritme)=SIN(2*pi/33*i) [return]
a(ritme)=(f(ritme)+g(ritme)+v(ritme))
/3 [return]
NEXT [return]

tekenen: [return]
SCREEN 2,640,216,3,2 [return]
WINDOW 2,"graphic",,0,2 [return]
CLS [return]
PALETTE 0,0,0,0 [return]
PALETTE 1,1,1,0 [return]
PALETTE 2,1,0,0 [return]
PALETTE 3,0,0,1 [return]
PALETTE 4,1,1,1 [return]
PALETTE 5,1,0,1 [return]
PALETTE 6,0,1,0 [return]
COLOR 4,0 [return]
PRINT "Bioritme voor : ";n$ [return]
LOCATE 1,40:PRINT "Geboren : ";gd;m$
(gm);gj [return]
dag1$=" 1   3   5   7   9  11  13  15
17  19  21  23  25  27  29  31" [return]
dag2$="   2   4   6   8  10  12  14  16
18  20  22  24  26  28  30  " [return]
ma=tm(m):IF m=2 AND j/4=INT(j/4) THEN
ma=ma+1 [return]
dag1$=LEFT$(dag1$,ma*2+1) [return]
dag2$=LEFT$(dag2$,ma*2+1) [return]
LOCATE 20,5:PRINT dag1$ [return]
LOCATE 21,5:PRINT dag2$ [return]
LOCATE 23,5:PRINT m$(m);j [return]
```

```
LINE (40,20)-(40,150),4 [return]
LINE (41,20)-(41,150),4 [return]
lad=ma*16+40 [return]
LINE (40,85)-(lad,85),4 [return]
LINE (40,86)-(lad,86),4 [return]
FOR dag=1 TO ma [return]
LINE (40+dag*16,20)-(40+dag*16,150),4
[return]
NEXT dag [return]
LOCATE 3,3:PRINT "+1" [return]
LOCATE 11,4:PRINT "0" [return]
LOCATE 19,3:PRINT "-1" [return]
COLOR 2,0 [return]
LOCATE 4,70:PRINT "Fysiek" [return]
COLOR 3,0 [return]
LOCATE 9,70:PRINT "Gevoel" [return]
COLOR 5,0 [return]
LOCATE 14,70:PRINT "Intellect" [return]
COLOR 1,0 [return]
LOCATE 19,70:PRINT "Gemiddeld" [return]
LINE (48,-f(1)*65+85)-STEP (0,0),2 [return]
LINE (48,-g(1)*65+85)-STEP (0,0),3 [return]
LINE (48,-v(1)*65+85)-STEP (0,0),5 [return]
LINE (48,-a(1)*65+85)-STEP (0,0),1 [return]
FOR dag=2 TO ma [return]
LINE (32+(dag-1)*16,-f(dag-1)*65+85)-(32+
dag*16,-f(dag)*65+85),2 [return]
LINE (32+(dag-1)*16,-g(dag-1)*65+85)-(32+
dag*16,-g(dag)*65+85),3 [return]
LINE (32+(dag-1)*16,-v(dag-1)*65+85)-(32+
dag*16,-v(dag)*65+85),5 [return]
LINE (32+(dag-1)*16,-a(dag-1)*65+85)-(32+
dag*16,-a(dag)*65+85),1 [return]
NEXT dag [return]
COLOR 6,0 [return]
LOCATE 25,1:PRINT "[+] maand verder
[-] maand terug    [enter] nieuwe bioritme

[*] end"; [return]

select: [return]
toets$=INKEY$ [return]
IF toets$="+" THEN m=m+1:GOTO vervolg
[return]
IF toets$="-" THEN m=m-1:GOTO vervolg
[return]
IF toets$=CHR$(13) THEN RUN [return]
IF toets$<>"*"THEN select [return]
WINDOW 1 [return]
COLOR 1,0 [return]
CLS [return]
END [return]

vervolg: [return]
IF m=13 THEN m=1:j=j+1 [return]
IF m=0 THEN m=12:j=j-1 [return]
GOTO berekening [return]
```

AmigaWaveCreator

Het volgende programma, van Maarten Hoeben maakt het mogelijk om zelf een golfvorm samen te stellen en deze in je eigen programma's te verwerken. In het volgende programma zitten een aantal standaard formules ingebouwd, zoals een zaagtand, sinus, puls golf e.d. Deze golfvormen zijn ook te kombineren. Achter in het programma staat een subroutine die men in de eigen programma's kan gebruiken om de golfvormen in te laden en af te spelen. Het gehele programma is menu gestuurd, waardoor het eenvoudig te gebruiken c.q uit te breiden is.

```
REM    >> Amiga Wave Creator << [return]
REM   [return]
REM    door:Maarten Hoeben [return]
REM          ir.m.schefferlaan 37 [return]
REM          7556CP  Hengelo (o) [return]

GOTO main: [return]

setup: [return]
GOSUB screeninit [return]
GOSUB varinit [return]

menuinit: [return]
MENU OFF [return]
MENU 1,0,1,"project" [return]
GOSUB projectmenu [return]
GOSUB wavemenu [return]
MENU 3,0,1,"golfvorm " [return]
GOSUB waveformmenu: [return]
MENU 4,0,1,"afspelen " [return]
GOSUB playwavemenu: [return]
MENU ON [return]
RETURN [return]

projectmenu: [return]
MENU 1,1,1,"veld schoon    " [return]
MENU 1,2,1,"herteken veld  " [return]
MENU 1,3,1,"golven schoon  "  [return]
MENU 1,4,1,"open           " [return]
MENU 1,5,1,"wegschrijven   " [return]
MENU 1,6,1,"stoppen        " [return]
RETURN [return]

wavemenu: [return]
MENU 2,0,1,"kies golf  " [return]
FOR t = 1 TO 3 : swave(t) = 1:NEXT [return]
swave(swave)=2 [return]
MENU 2,1,swave(1),"  golf 1     " [return]
MENU 2,2,swave(2),"  golf 2     " [return]
MENU 2,3,swave(3),"  golf 3     " [return]
RETURN [return]

waveformmenu: [return]
MENU 3,1,1,"sinus                " [return]
MENU 3,2,1,"zaagtand             " [return]
MENU 3,3,1,"driehoek golf        " [return]
MENU 3,4,1,"puls golf            " [return]
MENU 3,5,1,"willekeurige golf    " [return]
MENU 3,6,1,"geen golf            " [return]
MENU 3,7,1,"teken zelf een golf" [return]
MENU 3,8,1,"" [return]
MENU 3,9,1,"gemiddelde golf      " [return]
RETURN [return]

playwavemenu: [return]
MENU 4,1,1,"speel toon " [return]
RETURN [return]

screeninit: [return]
SCREEN 2,640,200,3,2 [return]
WINDOW 2,"Amiga Wave Creator",,7,2 [return]
paletteinit: [return]
col!=.4:FOR pal% = 4 TO 7 :col!=col!+.1
[return]
PALETTE pal%,col!,col!,col!:NEXT:RETURN
[return]

fieldinit: [return]
```

```
COLOR 1,0 [return]
AREA (0,0):AREA STEP (276,0) [return]
AREA STEP (0,138):AREA STEP (-276,0)
[return]
AREAFILL:COLOR 4,1 [return]
FOR t% = 0 TO 16:LINE (16,8+t%*8)-(272,8
+t%*8),6 [return]
LINE (t%*16+16,8)-(t%*16+16,135),6 [return]
LOCATE 1,(t%*2)+1:PRINT HEX$(t%) [return]
LOCATE t%+1,1 [return]
PRINT HEX$(t%) [return]
NEXT:RETURN [return]

fieldinit2: [return]
AREA (280,1):AREA STEP (330,0) [return]
AREA STEP (0,136):AREA STEP
    (-330,0)[return]
COLOR 1,1:AREAFILL:COLOR 3,1 [return]
LOCATE 3,47:PRINT"Amiga Wave Creator"
[return]
LOCATE 5,46:PRINT "door  Maarten
Hoeben" [return]
LINE (290,60)-(600,60),5:LINE (600,60)-
(600,130),5 [return]
LINE (600,130)-(290,130),5:LINE (290,
130)-(290,60),5 [return]
COLOR 0,1:RETURN   [return]

varinit: [return]
DEFINT c,t,W:DIM timbre(255),timbres(3,255
),swave(255),col(3), usevar(255) [return]
swave=1:col(1)=0:col(2)=2:col(3)=3:RETURN
[return]

varclear: [return]
FOR t = 0 TO 255 : FOR k = 1 TO 3 [return]
timbres(k,t)=0:NEXT:NEXT:GOSUB fieldinit:
RETURN [return]

main: [return]
GOSUB setup [return]
GOSUB fieldinit:GOSUB fieldinit2 [return]

menucontrol: [return]
menuid = MENU (0):menuitem = MENU (1)
[return]
ON menuid GOSUB project,waveselect,wave
form,playmenu [return]
GOTO menucontrol [return]

project: [return]
ON menuitem GOSUB fieldinit,redraw,var
clear,openwave,savewave,quit
RETURN [return]

redraw: [return]
GOSUB fieldinit:store=swave:FOR swave =
1 TO 3 [return]
GOSUB drawwave:NEXT:swave=store:RETURN
[return]

namewave: [return]
GOSUB fieldinit2:LOCATE 11,40 [return]
INPUT "Naam:";wavename$ [return]
RETURN [return]

openwave: [return]
GOSUB fieldinit2:LOCATE 9,40 [return]
PRINT "Golf laden":GOSUB playwaveselect
[return]
GOSUB namewave:LOCATE 13,40 [return]
PRINT "Loading "wavename$" in golf"swave
[return]
OPEN wavename$ FOR INPUT AS #1:t%=0
[return]
WHILE NOT EOF(1) [return]
INPUT #1,timbres(swave,t%):t%=t%+1 [return]
WEND:CLOSE #1:GOSUB drawwave [return]
GOSUB fieldinit2:LOCATE 9,40:PRINT "Ok,
gedaan." [return]
RETURN [return]

savewave: [return]
GOSUB fieldinit2:LOCATE 9,40 [return]
PRINT "Golf wegschrijven":GOSUB
playwaveselect [return]
GOSUB namewave:LOCATE 13,40 [return]
PRINT "Saving golf"swave" als
"wavename$ [return]
FOR t% = 0 TO 255:usevar(t%)=INT
(timbres(swave,t%)):NEXT [return]
OPEN wavename$ FOR OUTPUT AS #1 [return]
FOR t% = 0 TO 255 : WRITE #1,usevar(t%)
[return]
NEXT : CLOSE #1 [return]
GOSUB fieldinit2:LOCATE 9,40:PRINT "Ok,
gedaan." [return]
RETURN [return]

quit: [return]
SCREEN CLOSE 2 [return]
MENU RESET [return]
CLS:END [return]

waveselect: [return]
swave=menuitem:GOSUB wavemenu:RETURN
[return]

drawwave: [return]
linkvar%=(timbres(swave,1)/2):
FOR t% = 0 TO 255  [return]
LINE (t%+16,72+linkvar%)-(t%+17,72+
(timbres(swave,t%)/2)),col(swave)
linkvar%=(timbres(swave,t%)/2):
NEXT:RETURN [return]

waveform: [return]
GOSUB fieldinit2 [return]
ON menuitem GOSUB fr1,fr2,fr3,fr4,fr5,
fr6,fr7,fr8,fr8 [return]
GOTO drawwave [return]

varinput: [return]
LOCATE 11,40:INPUT "Begin (0-255)";
start% [return]
LOCATE 12,40:INPUT "Einde (0-255)";
einde% [return]
LOCATE 13,40:INPUT "V1    (0-100)";
v1! [return]
LOCATE 14,40:INPUT "V2    (0-255)";
v2! [return]
LOCATE 15,40:INPUT "V3    (0-127)";
v3! [return]
RETURN [return]

fr1: [return]
LOCATE 9,40:PRINT "Sinus: Y=sin((X*V1)
+V2)*V3" [return]
GOSUB varinput [return]
FOR t% = start% TO einde% [return]
timbres(swave,t%)=SIN(((t%/40.9)*v1)
+v2)*v3 [return]
```

```
NEXT:RETURN [return]

fr2: [return]
LOCATE 9,40:PRINT "Zaagtand: V1=breedte
V2=plaats" [return]
LOCATE 10,50:PRINT "V3=hoogte" [return]
GOSUB varinput [return]
ste!=(v3/2)/v1:amp!=1*v2:FOR t%=0 TO
255 [return]
usevar(t%)=amp!:amp=amp+ste! [return]
IF amp>v3 THEN amp=-v3 [return]
NEXT:FOR t%=start% TO einde% [return]
timbres(swave,t%)=usevar(t%):NEXT [return]
RETURN [return]

fr3: [return]
LOCATE 9,40:PRINT "Drieh. golf: V1=breed
te V2=plaats" [return]
LOCATE 10,53:PRINT "V3=hoogte" [return]
GOSUB varinput [return]
ste!=(v3/2)/v1:amp!=1*v2:FOR t%=0 TO
255 [return]
usevar(t%)=amp!:amp=amp+ste! [return]
IF amp>v3-2 OR amp<-v3+2 THEN ste!=
-ste! [return]
NEXT:FOR t%=start% TO einde% [return]
timbres(swave,t%)=usevar(t%):NEXT [return]
RETURN [return]

fr4: [return]
LOCATE 9,40:PRINT "Puls golf:  V1=breed
te V2=breedte 2" [return]
LOCATE 10,52:PRINT "V3=hoogte" [return]
GOSUB varinput [return]
ste!=v1:FOR t% = 0 TO 255:usevar(t%)=v3:
ste!=ste!-1 [return]
IF ste!=<0 AND v3=>0 THEN v3=-v3:ste!=
v1 [return]
IF ste!=<0 AND v3=<0 THEN v3=-v3:ste!=
v2 [return]
NEXT:FOR t%=start% TO einde% [return]
timbres(swave,t%)=usevar(t%):NEXT [return]
RETURN [return]

fr5: [return]
LOCATE 9,40:PRINT "Willek. golf:V1=plaats
    V2
=geen nut" [return]
LOCATE 10,53:PRINT "V3=hoogte" [return]
GOSUB varinput [return]
FOR t% = 0 TO 255 [return]
timbres(swave,t%) = INT ((RND(1)*v3*2)+v1)-
127 [return]
NEXT:RETURN [return]

fr6: [return]
LOCATE 9,40:PRINT "Geen golf:" [return]
FOR t% = 0 TO 255:timbres(swave,t%)=0:
NEXT:RETURN [return]

fr7: [return]
LOCATE 9,40:PRINT "Teken golf:" [return]
LOCATE 11,40:PRINT "U kunt zelf een golf
vorm tekenen." [return]
LOCATE 12,40:PRINT "Om te starten moet u
de linker muis" [return]
LOCATE 13,40:PRINT "toets indrukken.Ver
volgens kunt u " [return]
LOCATE 14,40:PRINT "muis op en neer be
wegen om de golf" [return]
LOCATE 15,40:PRINT "te tekenen.Stop
door op de spatie " [return]
LOCATE 16,40:PRINT "te drukken." [return]
x=MOUSE(0) [return]

markstart: [return]
IF MOUSE(0)=-1 THEN x = MOUSE (1) :
y = MOUSE (2) [return]
IF x>15 AND x<273 AND y>7 AND y<
137 THEN buttoncontrol [return]
GOTO markstart [return]

buttoncontrol: [return]
IF MOUSE (0) = -1 THEN mousepos [return]
IF INKEY$ = " "THEN fieldinit [return]
GOTO buttoncontrol [return]
mousepos: [return]
x=x+1:y=MOUSE(6) [return]
IF x>271 THEN x=x-1 [return]
IF y>8 AND y<136 THEN GOSUB drawpoint
[return]
GOTO buttoncontrol [return]

drawpoint: [return]
LINE(x,y)-(x,y),col(swave) [return]
timbres(swave,(x-16))=(y*2)-144:RETURN
[return]

fr8: [return]
LOCATE 9,40:PRINT"Gemiddelde golf:"
[return]
LOCATE 11,40:PRINT"Van alle drie de gol
ven wordt het" [return]
LOCATE 12,40:PRINT"gemiddelde genomen.
Dat gemiddelde" [return]
LOCATE 13,40:PRINT"wordt dan in golf 1 ge
zet." [return]
LOCATE 14,40:INPUT"Begin (0-255)";start%
[return]
LOCATE 15,40:INPUT"end   (0-255)";einde%
[return]
FOR t% = start% TO einde% [return]
timbres (1,t%)=(timbres(1,t%)+timbres
(2,t%)+timbres(3,t%))/3
NEXT:swave=1:GOSUB wavemenu:GOSUB
fieldinit:RETURN [return]

playwaveselect: [return]
LOCATE 11,40:PRINT"0 = Huidige ge
bruikte golf" [return]
LOCATE 12,40:PRINT"(1-3) kies golf"
[return]
LOCATE 14,40:INPUT"golf (1-3)";store
[return]
IF store = 0 THEN RETURN [return]
swave = store:GOSUB wavemenu:RETURN
[return]

frequention: [return]
GOSUB fieldinit2:LOCATE 9,40:PRINT "Fre
quentie:" [return]
LOCATE 11,40:INPUT"Herz :";herz [return]
IF herz = 0 THEN herz = 440 [return]
RETURN [return]

playmenu: [return]
ON menuitem GOSUB playwave [return]
RETURN [return]

playwave: [return]
GOSUB fieldinit2:LOCATE 9,40:PRINT"
Speel toon:" [return]
```

```
GOSUB playwaveselect:GOSUB frequen
tion [return]
GOSUB fieldinit2:LOCATE 12,40:PRINT"
druk op de spatie " [return]
FOR t% = 0 TO 255:timbre(t%)=timbres
(swave,t%):NEXT [return]
WAVE 0,timbre [return]
soundcont: [return]
SOUND herz,20,100,0 [return]
IF INKEY$=" "THEN GOSUB fieldinit2:
RETURN [return]
GOTO soundcont [return]

REM  De volgende subroutine kan worden
gebruikt om de weggeschreven [return]
REM  geluidsgolven in uw eigen programma
     te
gebruiken.Voor aan het [return]
REM  programma hoort dan de 1e regel te
staan.Hierin wordt timbre  [return]
REM  gedimensioneerd. [return]

' DIM timbre(255) [return]

' OPEN wavename$ FOR INPUT AS #1:t%=0
[return]
' WHILE NOT EOF(1) [return]
' INPUT #1,timbre(t%):t%=t%+1 [return]
' WEND:CLOSE #1 [return]

' WAVE (0-3),timbre [return]
```

prijsvraag

Het is ons dan toch gelukt, om uit de stapels inzendingen die programma's te kiezen die de prijzen zouden winnen. Weken heb ik niets anders kunnen doen dan de programma's bekijken. Niet alleen het aantal inzendingen was dit jaar hoger dan anders, ook de kwaliteit lag beduidend hoger dan vorige jaren.
Voor het beoordelen van de programma's hebben we een cijfer gegeven van 1 tot 10. Na afloop bleek maar liefst dat 72 % van de ingezonden programma's met een voldoende te zijn beoordeeld. Hiervan kreeg 35 % een cijfer van 7 en hoger. Uit het aantal inzendingen uit Belgie, 25 % kwam hier vandaan, blijkt dat ons blad steeds internationaler wordt.

De eerste prijs is gewonnen door **R. van de Klippe** *uit Lemelerveld. Hij krijgt hiervoor een kleuren monitor.*

Voor de tweede prijs komt een programmeur in aanmerking, die graag met de C 16 werkt: **Theo Wiersma** *uit Zwijndrecht.*

De derde prijs is een printer en die komt bij een inzender waar we al een aantal keren wat van geplaatst hebben: **Y. Rozijn** *uit Amsterdam*

De volgende inzenders zien hun programma bekroond met een mooi software/hardware pakket:

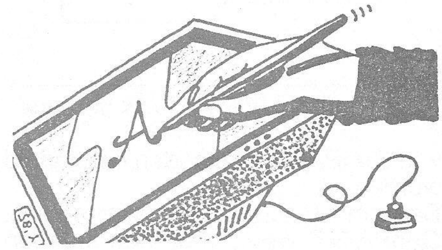
-S. Jokhan uit Surhuisterveen
-Dirk Bouma uit Nijland
-P van de Mespel uit Den Haa
-Pieter de Wijn uit Harelbeke Belgie
-J.W. Wijmwa uit Kootstertille
-Remy de Ruysscher uit Bunnik
-A.P.W. Baelde uit Maarsen
-M. Geurts uit Horst
-W. Stadhouders uit Terneuzen
-J. Schipper uit Bunnik

Voor de kategorie hardware zijn een aantal leuke ideeën naar voren gekomen. We bekijken nu welke we hiervan, in samenwerking met de maker, kunnen gebruiken voor een artikel.
Dus de makers van programma's in deze kategorie zullen nog even geduld moeten hebben, alvorens de uiteindelijke resultaten bekend worden.

Rob Goudriaan.

GEOS, het operating system voor de C-64/128 met icons, pulldown menus en WYSIWYG is nu een jaar oud. Alom geprezen en meermalen onderscheiden met verschillende prijzen staat het nog steeds in de belangstelling. Tot nu toe is echter weinig bekend over de manier waarop GEOS werkt. Peter Boncz gaat hier in op de basis van de kracht van GEOS: het VLIR file-systeem. Bij dit artikel staat een listing om GeoPAINT-tekeningen in (BASIC) programma's te gebruiken.

Data-opslag in GEOPAINT



Het tekenprogramma Geopaint steekt met kop en schouders boven al zijn konkurrenten uit omdat het een combinatie biedt van bedieningsgemak, uitgebalanceerde schermopmaak en vooral een grote veelzijdigheid. Die veelzijdigheid komt tot uiting in de tientallen verschillende functies, die het pakket in zich heeft. Nadelen, die natuurlijk ook genoemd moeten worden, zijn de traagheid door het vele laden van disk, en de krappe diskruimte die overblijft voor bestanden van de gebruiker.

Het geheim achter de veelzijdigheid van GeoPAINT is dat het komplete programma helemaal niet in een C-64 past, er is steeds maar een gedeelte aanwezig. Als een functie wordt aangeroepen, leest GEOS het gewenste deel van disk. Hier duikt dan ook de VLIR-structuur op. (**VLIR** = Variable Lenght Indexed Records). GeoPAINT is in stukjes gehakt, en naast het basisgedeelte worden steeds verschillende brokjes ingeladen. GeoPAINT is dus, net als alle andere VLIR-files een verzameling bestanden.
De VLIR structuur wijkt van de standaard 1541-diskformat af op de volgende punten:

- ° de directory.
- ° de verdeling van een file in losse stukjes (Index table).
- ° extra gegevens over het bestand, geplaatst in een 'fileheader'.
- ° de wegschrijf-code (deze is niet ASCII).

Directory entry

De 1541 diskdrive formateert een disk in 35 tracks. Deze zijn op hun beurt weer onderverdeeld in sectoren. In track 18 is de directory gehuisvest. In een directory-sector staan maximaal 8 bestanden vermeld, ieder bestand heeft 32 bytes. Deze 32 bytes worden de Directory Entry genoemd.
De eerste twee bytes van een sector geven altijd aan op welke track en sector het programma (of de directory) waar die sector een deeltje van is, verdergaat.
Na deze twee bytes begint de Directory Entry. Byte 0 (gerekend na die eertse twee bytes) geeft het type file weer (PRG SEQ etc.) Byte 1 en 2 geven de start track en sector van het bestand, in Byte 3 en verder volgt de naam in ASCII. Na 32 bytes herhaalt zich dit patroon, dus in byte 33 en 34 staan de start track en sector van de volgende file, gevolgd door de naam etc. (zie illustratie 1).

	DIRECTORY ENTRY (Also Known as File Entry)
0	c64 File Type: 0=DELeted, 1=SEQuential, 2=PRoGram, 3=USeR, 4=RELative. Bit 6=Used as Write-Protect Bit.
1	Track and Sector of First Data Block in This File.
2	If the File Is VLIR then this Word Points to the Index Table Block.
3 ... 18	16 Character File Name Padded with Shift Spaces $A0
19 - 20	Track and Sector of GEOS File Header (New Structure)
21	GEOS File Structure Type: 0=SEQuential, 1=VLIR
22	GEOS File Types: 0=NOT_GEOS, 1=BASIC, 2=ASSEMBLY, 3=DATA, 4=SYSTEM, 5=DESK_ACC, 6=APPLICATION, 7=APPL_DATA, 8=FONT, 9=PRINTER, 10=INPUT_DEVICE, 11=DISK_DEVICE, 12=SYSTEM_BOOT, 13=TEMPORARY (for Swap Files)
23	Date: Year Last Modified, Offset from 1900
24	Date: Month Last Modified (1-12)
25	Date: Day Last Modified (1-31)
26	Date: Hour Last Modified (0-23)
27	Date: Minute Last Modified (0-59)
28 - 29	Low Byte, High Byte: Number of Blocks (Sectors) in the File

Illustratie 1.

Index Table

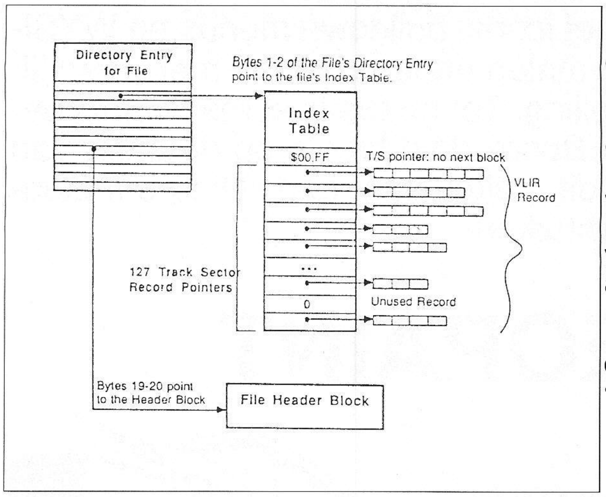
De directory-entry geeft in Byte 1 en 2 de track en sector van een zg. Index Table (ook wel subdirectory) aan. Deze Index Table bestaat uit 1 sector. In Index Table worden alle losse stukjes file (Records) aan elkaar geknoopt. De eerste twee bytes in de Index Table hebben de waardes 0 en $ff. Hierna beginnen we te tellen. Byte 0 en 1 geven respectievelijk de track en sector van Record 1, Byte 2 en 3 die van Record 2, enzovoort tot Byte 256 (einde sector). (illustratie 2)

Fileheader

In de VLIR-structuur is tot slot nog de fileheader belangrijk. Deze bestaat evenals de Index Table uit 1 sector. Ieder GEOS bestand heeft een fileheader. Daarin staat informatie over het startadres van de routine, data van het icon, de geheugenlocatie van de routine, bij GET-INFO ingetikte tekst, etc. (zie ill. 3). De track en sector van de fileheader staan in de Directory Entry aangegeven na de naam, dus op byte 19 en 20. (zie ill. 1)

CBM-graphics

Omdat er nu is uitgelegd, hoe GEOS-applicaties, zoals GeoPAINT en GeoWRITE, hun data op disk zetten, kan er dieper ingegaan worden op de specifieke eigenschappen van GeoPAINT. Dan kan ook de CODERING



Ill. 2 De VLIR structuur

van GeoPAINT bestanden behandeld worden.
Daarom nu terug naar de graphics. GeoPAINT werkt met een bit-map mode HIRES scherm. (*d.w.z. 25x40 blokjes van 8x8 puntjes, ieder blokje heeft max. twee kleuren*). Het nadeel van deze HIRES-mode is dat de mogelijkheden om met kleuren te werken beperkter zijn als die van de multicolor mode. Het grote voordeel is, dat hiermee een tweemaal hogere resolutie bereikt wordt. Het is echter noodzakelijk dat GEOS-applicaties met een bit-mapped scherm werken, de standaard (machinetaal)routines voor pull-down menu's en tekenfuncties kennen namelijk alleen deze mode.

Het bit-mapped beeldscherm is ingedeeld in CARDS (blokjes). CARD 1 staat in de linkerbovenhoek, de volgende staat rechts daarvan, tot en met CARD 40 in de rechterbovenhoek. Dan gaat het een regel naar onderen. (zo door tot en met CARD 1000). Omdat iedere CARD een raster van 8x8 is, bestaat hij uit 8 bytes. Het complete scherm bestaat dus uit 8000 bytes. (ill. 4.)

Het tekenvel van GeoPAINT bestaat uit 80 CARDS horizontaal en 90 CARDS vertikaal. (dus 640x720 puntjes). Alle GEOS files, dus ook tekeningen, worden volgens het VLIR systeem op disk gezet. Een GeoPAINT tekening bestaat dus uit stukjes, hij is in stroken gesneden. Ieder stukje (Record) bestaat uit twee horizontale regels van 80 CARD, dus 2x80x8=1280 bytes. Na iedere strook volgt de color table, waarin de gegevens over de kleur van de strook zijn opgeslagen. De kleur wordt per CARD bepaald (zie los kader), dus 2x80=160 bytes voor de color table.

Codering

Om ruimte op disk te sparen, comprimeert (verkort) GeoPAINT alle tekeningen. Er zijn daarom twee soorten data: data die de wijze van opslaan (count mode) aangeven, en data van die de gegevens zelf aangeven.

Er bestaan drie count modes:

° 1) waarde x = 0-64. Functie: de volgende x bytes zijn niet verkort, ze zijn 'zichzelf'.

° 2) waarde x = 65-128. Functie: (x-64) maal het patroontje van de 8 daaropvolgende bytes.

° 3) waarde x = 129-255. Functie: (x-128) maal de volgende byte.

Om een voorbeeld te geven:

02, 01, 198, 66, 01, 03, 15, 31, 15, 07, 03, 01, 00, 255, 0

betekent:
Twee (02) losse waardes : 01,198
Tweemaal (66-64) het patroon : 01,03,15,31,15,03,01,00
127 maal (255-128) een 00

Aanzienlijke geheugen- en diskwinst kan dus worden geboekt als een groot deel van de tekening met een bepaald patroontje is gePAINT, of als er lege stukken in de tekening zitten. Een VLIR file neemt gemiddeld 20 tot 30 procent van de ruimte van een normaal bestand in!

Zelf sleutelen

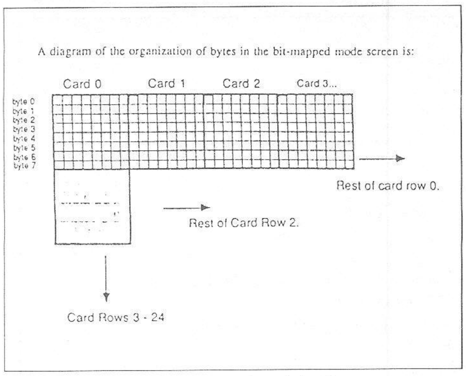
Nu we weten hoe een GeoPAINT-file in elkaar zit en hoe hij gecomprimeerd wordt, kunnen we zelf wat gaan sleutelen. Het is bijvoorbeeld mogelijk een SCREENCATCHER te schrijven. Dit is een programma, dat HIRES schermen in de besproken code omzet, en vervolgens in VLIR wegschrijft. Dit is

GEOS File Header Structure

(128 bytes. New GEOS file. Pointed to by Directory Entry)

Byte No	Contents	Description
0-1	00,FF	00 indicates this is the last block in the file. FF is the index to the last valid data byte in the block.
2	3	Width of icon in bytes, always 3
3	21	Height of file icon in lines, always 21.
4	$80 + 63	Bit Map data type. Top bit = 1 means the lower 7 bits contain the number of unique bytes which follow, i.e. 63. Always this value.
5 - 67	$FF,$FF,$FF ... $FF,$FF,$FF	Start of picture data total of 63 bytes used to define icon graphic End of picture data
68	$80 + PRG	C-64 file type, used when saving the file under GEOS PRG = 1, SEQ = 2, USR = 3, REL = 4. Bit 6 = 1 Write Protected.
69	$02	GEOS file type: BASIC = 1, ASSEMBLY = 2, DATA = 3, SYSTEM = 4, DESK_ACC = 5, APPLICATION = 6, APPL_DATA = 7, FONT = 8, PRINTER = 9, INPUT_DEVICE = 10, DISK_DEVICE = 11, SYSTEM_BOOT=12, TEMPORARY=13
70	0	GEOS structure type, 1 = VLIR, 0 = SEQUENTIAL
71 - 72	FileStart	Start address in memory for loading the program
73 - 74	FileEnd	End address in memory for loading the program
75 - 76	InitProg	Address of initialization routine to call after loading the program
77 - 96	Filename 0,1,..2,0,0,	20 byte ASCII application filename. Bytes 0-11 = the name padded with spaces; 12-15=version string "V1.3"; 16-20=0's
97 - 116	Parent Disk Author Name	If Data file, 20 byte ASCII filename of parent application's disk. If application program, holds name of software designer.
117 - 136	Parent Application	If Data file, 20 byte parent application filename. Bytes 0-11=name padded with spaces; 12-15=version string "V1.3"; 16-20=0's
137 - 159	Application	23 bytes for application use..
160 -255	Get Info	Used for the file menu option getInfo. String must be null terminated and null must be first char if string is empty.

Ill. 3. File Header structuur



Ill. 4. Bit-Map beeldscherm

door mij gedaan, maar dat programma is te uitgebreid om hier af te drukken. De bij deze gepubliceerde listing betreft een programma dat precies het omgekeerde doet: het leest VLIR files en zet de code om, zodat de tekening 'op ware grootte' in het HIRES geheugen verschijnt.

Dit programma schept grote perspectieven voor de programmeurs onder u. Als u altijd al vond dat uw BASIC programma's wat kaal aandeden, ligt hier uw kans. Met de ruime mogelijkheden van GeoPAINT kunnen hele mooie laadplaten, schema's e.d. gemaakt worden. De VLIR-IMPORTER leest ze vervolgens probleemloos voor u in.

Veel succes ermee!

P.Boncz

De VLIR IMPORTER

Het programma bestaat grofweg uit 3 delen:

- De directory-lezer. Na het intikken van de naam leest het programma waar de Index Table op disk staat.
- Vervolgens inlezen van die Index Table. Op deze manier komt het programma te weten waar de verschillende brokken van de tekening zich bevinden.
- Ten slotte worden deze brokken stuk voor stuk ingelezen, gecodeerd en in het grafisch geheugen gePOKEd.

Bijzonderheden:

Het gebruikte HIRES geheugen begint bij geheugenplaats 24576 ($6000) en eindigt bij 32576 ($7f40). Het kleurengeheugen wat daarbij hoort begint bij 17408 en eindigt op 18408. De waarde van het kleurengeheugen wordt als volgt bepaald: de kleur van de achtergrond waarbij opgeteld de kleur van de tekening maal 16. Om het scherm aan te zetten moeten de volgende POKE's worden verricht:
POKE 53265,187 : POKE 53272,31 : POKE 56576,2

De omgekeerde handeling is:
POKE 53265,155 : POKE 53272,23 : POKE 56576,3
Omdat het Geopaint tekenvel (90x80 CARDS) groter is dan een C-64 scherm (25x40 CARDS) importeerd dit programma de linkerbovenhoek van het tekenvel. Als het programma klaar is met importeren (dit kan wel zo'n 5 minuten duren!) kan het scherm als volgt gesaved worden:
POKE 44,68 : POKE 43,0 : POKE 46,127 : POKE 45,64
SAVE"naam",8,1
Met **LOAD"naam",8,1** wordt het scherm weer geladen.

Wie nog niet bekend is met de HIRES van de C-64 raad ik aan voor aanvullende informatie de eerste twee jaargangen van C-I nog eens door te bladeren. Het voert te ver om dat in dit artikel nog uitgebreid te bespreken.

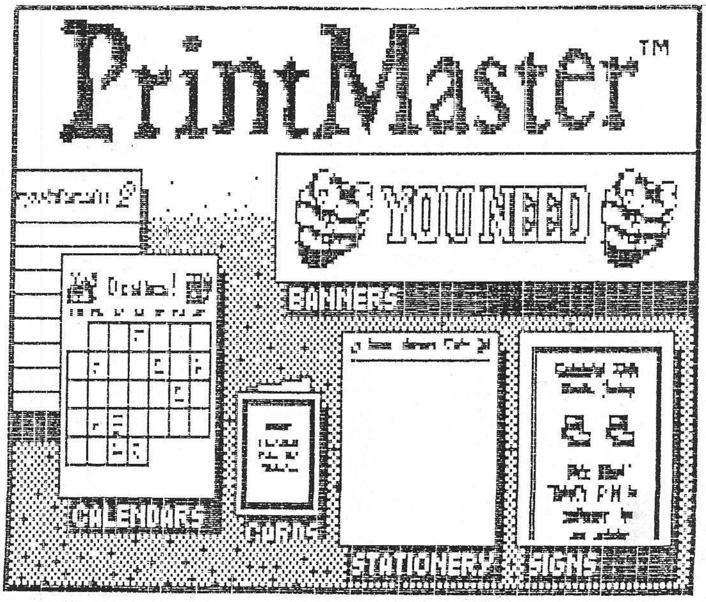
Listing VLIR Graphics Importer

```
100 rem
110 rem
120 rem ********************************
130 rem ***                          ***
140 rem *** vlir graphics importer ***
150 rem ***                          ***
160 rem ********************************
170 rem
180 rem
190 rem ********************
200 rem * dit programma *
210 rem * is geschreven *
220 rem *       door:       *
230 rem *  peter boncz  *
240 rem ********************
250 rem
260 rem
270 rem naam & vinden fileheader
280 rem
290 rem
300 dimn$(16):dimnn(17):dimns(17)
310 poke53280,0:poke53281,0:poke646,1
320 print"[CLR-HOME][5xneer]▩[7xspatie]
    in te lezen geopaint file:[4xneer]"
        330 input n$
340 poke53265,187:poke56576,2:poke53272
    ,31
350 open15,8,15
360 open5,8,5,"#"
370 vt=18:vs=1:t=0
380 print#15,"b-r:"5;0;vt;vs
390 print#15,"b-p:"5;0;
400 gosub1440:vt=by:gosub1440:vs=by
410 gosub1440:ib=by
420 ifib<>131then500
430 gosub1440:bt=by:gosub1440:bs=by
440 fora=1to16:gosub1440:ns(a)=by:nn(a)
    =by:ifby<123andby>96then nn(a)=by-3
    2
450 ifns(a)=160then470
460 nexta
470 n1$="":n2$=""
480 forb=1toa-1:n1$=n1$+chr$(ns(b)):n2$
    =n2$+chr$(nn(b)):next
490 ifn1$=n$orn2$=n$then560
500 t=t+1
510 ift=9then540
520 print#15,"b-p:"5;(t*32+2);:dt=(t*32
    )+2
530 goto410
540 if vt=0thenprint"naam bestaat niet"
    :goto1340
550 dt=0:t=0:goto 380
560 rem
570 rem
580 rem inlezen fileheader &
590 rem bepalen vlir-records
600 rem
610 rem
620 dt=0
630 print#15,"b-r:"5;0;bt;bs
640 print#15,"b-p:"5;0;
650 gosub1440:gosub1440
660 fort=1to13
670 gosub1440:poke50000+(t-1)*3,by:gosub
    1440:poke50001+(t-1)*3,by
680 ifpeek(50000+(t-1)*3)=0thenpoke5000
    2+(t-1)*3,1:goto700
690 poke50002+(t-1)*3,0
```

vervolg VLIR-listing

```
700 next t
710 close5:close15
720 rem
730 rem
740 rem inlezen vlir-records
750 rem
760 rem
770 t=1
780 ift=14then1350
790 poke50040,t:clr
800 t=peek(50040):bt=peek(50000+(t-1)*3
    ):bs=peek(50001+(t-1)*3)
810 bm=peek(50002+(t-1)*3):tt=0
820 ifbm=1thengoto1700
830 vt=bt:vs=bs:dt=0
840 open15,8,15:open5,8,5,"#"
850 print#15,"b-r:"5;0;vt;vs
860 print#15,"b-p:"5;0;
870 gosub1440:vt=by:gosub1440:vs=by
880 rem
890 rem
900 rem bepalen count-mode
910 rem
920 rem
930 gosub1440:ko=by
940 ifko<64thengoto1110
950 ifko<128thengoto1220
960 rem
970 rem
980 rem count-mode 1
990 rem
1000 rem
1010 gosub1440
1020 fora=1toko-128
1030 gosub1570
1040 nexta:iftt<1441then910
1050 t=t+1:goto780
1060 rem
1070 rem
1080 rem count-mode 2
1090 rem
1100 rem
1110 ei=ko
1120 fora=1toei
1130 gosub1440
1140 gosub1570
1150 nexta:iftt<1441then910
1160 t=t+1:goto780
1170 rem
1180 rem
1190 rem count-mode 3
1200 rem
1210 rem
1220 ei=ko-64:b=1
1230 ifb>8then1260
1240 gosub1440:ns(b)=by:b=b+1
1250 goto1230
1260 b=1:a=1
1270 ifb>eithen1310
1280 by=ns(a):a=a+1:gosub1570
1290 if a=9thena=1:b=b+1:goto1270
1300 goto1280
1310 ift<1441then910
1320 t=t+1:goto780
1330 poke198,0:wait198,1
1340 close5
1350 print#15,"i"
1360 close15
1370 poke53265,155:poke56576,3:poke53272
    ,23
1380 end
1390 rem
1400 rem
1410 rem 'get' routine
1420 rem
1430 rem
1440 print#15,"b-p:"5;dt;:get#5,by$:by$=
     by$+chr$(0):dt=dt+1
1450 ifby$=chr$(0)thenby=0:goto1470
1460 by=asc(by$)
1470 ifdt=256thengoto1490
1480 return
1490 dt=2:print#15,"b-r:"5;0;vt;vs:print#
     15,"b-p:"5;0;:get#5,by$
1500 vt=asc(by$+chr$(0)):print#15,"b-p:"
     5;1;:get#5,by$:vs=asc(by$+chr$(0))
1510 goto1480
1520 rem
1530 rem
1540 rem 'poke' routine
1550 rem
1560 rem
1570 iftt<320thenpoke24576+(t-1)*640+tt,
     by:tt=tt+1:return
1580 iftt<640thentt=tt+1:return
1590 iftt<960thenta=tt-320:poke24576+(t-
     1)*640+ta,by:tt=tt+1:ta=tt:return
1600 iftt<1288thentt=tt+1:return
1610 iftt<1328thenta=tt-1288:poke17408+(
     t-1)*80+ta,by:tt=tt+1:return
1620 iftt<1368thentt=tt+1:return
1630 iftt<1408thenta=tt-1368:poke17448+(
     t-1)*80+ta,by:tt=tt+1:return
1640 tt=tt+1:return
1650 rem
1660 rem
1670 rem 'poke leeg' routine
1680 rem
1690 rem
1700 fora=24576+(t-1)*640toa+639:pokea,0
     :nexta:fora=17408+(t-1)*80toa+79
1710 pokea,191:nexta:t=t+1:goto780

*** EINDE LISTING ***
```



Amiga

Upgrade Music Studio

In Amerika heeft Activision de versie 2.0 van The Music Studio aangekondigd, waarvan geregistreerde gebruikers van The Music Studio (Amiga of Apple II) een upgrade kunnen krijgen. Het programma werkt op Amiga's met minimaal 512 K RAM en Kickstart 1.2. The Music Studio is een zeer uitgebreid programma voor het werken met MIDI-apparatuur.

DeluxeProductions

In de Deluxe serie van Electronic Arts is Deluxe Productions voor de Amiga voor professioneel videoproducties en zakelijke presentaties. Het programma gebruikt 16 kleuren en hoge-resolutie graphics. Er zijn vele editing technieken en speciale effecten. Het programma kan met alle andere Deluxe programma's samenwerken.

F-18 Interceptor

Ook van Electronic Arts komt de zeer geavanceerde flight-simulator voor de Amiga. Er kan gekozen worden tussen de F-18 Hornet en de F-16 Falcon, beide in werkelijheid ook bestaande gevechtsvliegtuigen. Men kan op verschillende vliegvelden of een vliegdekschip starten en landen. Er zijn zes verschillende misssies, zoals het beschermen van het vliegtuig van de president of het voorkomen van wereldoorlog drie. Er is een zeer gedetailleerde grafische afbeelding van San Fransisco in het spel ingebouwd.

Blackshadow

Dit spel voor de Amiga heeft niets met oogschaduw te maken maar is een grafisch fraai spel waarin de aarde wordt bedreigd door een langzaam naderbij komende astroïde, die wordt beschermd door een krachtveld waar geen mens doorheen kan breken. De speler wordt nu als vrijwilliger naar deze ster gestuurd, omdat wetenschappers een methode hebben gevonden om het krachtveld kort te onderbreken. Aan de speler om de aarde te redden van de dreiging die Blackshadow vormt.

From Russia

Het eerste Russische computerspel dat in het westen is doorgedrongen heet Tetris en wordt door Mirrorsoft op de markt gebracht. Het spel werd bedacht door een onderzoeker van de Academie der Wetenschappers in Moskou en geprogrammeerd door een student van de Moskouse universiteit. Tetris is een eenvoudig maar zeer verslavend abstract puzzelspel, beschikbaar voor C-64, Amiga, Atari ST en IBM-compatibles.

Professor Eyestrain

Terramexis een arcade-spel waarin Professor Albert Eyestrain waarschuwt tegen een atroide die op een aanvaringskoers met de aarde is (waar hebben we dat eerder gezien). Albert is in z'n jeugd al eens van school gestuurd omdat hij de auto van het hoofd van de school in boter had veranderd. Nu wordt hij dan ook niet geloofd, totdat het bijna te laat is. De professor is dan al kniezend verdwenen, en de speler wordt geacht hem weer op te sporen voordat het echt te laat is. Daarna moet Eyestrain ook nog overreed worden om de wereld te redden. Geen geringe taak, die met de C-64, Amiga of Atari ST kan worden uitgevoerd.

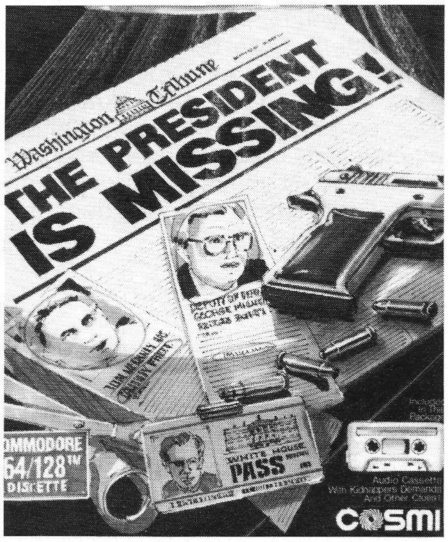
C-64

Majong

Een computerversie van deze chinese methode om te jokeren is gemaakt door CDS software (engeland) volgens regels die zijn aanbevolen door de Britse Mah-Jong vereniging. Op het beeldscherm speelt je hand in 3-D graphics tegen de computer. Er is een demonstratiemogelijkheid ingebouwd en er zijn vier oefensessies mogelijk. De computer kan op negen niveaus van intelligentie worden ingesteld. Er komen naast de C-64 cassette en diskette ook versies voor de Amiga, Atari ST en DOS computers.

The President is missing

Bij dit spel voor de C-64 zijn een aantal mysterieuze aanwijzingen op een audio-cassette bijgevoegd. Daarmee moeten we proberen de president van de Verenigde Staten en andere deelnemers aan een geheime economische topconferentie uit de handen van zijn ontvoerders halen. Hulpmiddel is verder de centrale computer van de contra-terrorisme afdeling van de CIA. Als je doordingt tot het geheim van de ontvoering zal echter blijken dat daarachter een nog veel wilder en sinister plan schuilgaat.

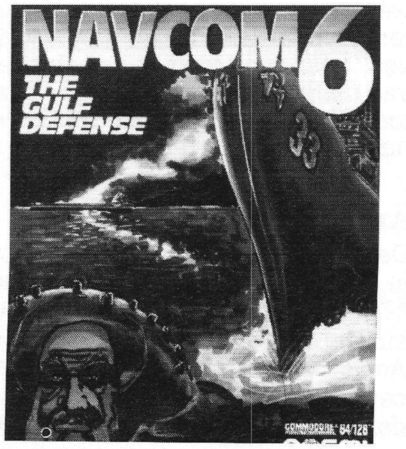


Road Warrior

Als je houdt van joy-rijden in een auto of op een motor dan is dit het juiste spel. Er wordt in aangezet tot haat tegens Hells Angels, want die horen volgens de begeleidende brief niet op de weg. Zoals zoveel andere 'spelletjes' speelt ook hierin geweld een belangrijke rol. Onder begeleiding van muziek en met behulp van verschillende wapens kun je in je Ferrari of Volkswagen de straten schoonmaken.

Gulf Defense

In navolging van de Nederlandse marineschepen kan men ook met de Commodore 64 of 128 naar de Golf gaan om tankers te escorteren, mijnen op te ruimen en je te beschermen tegen raids van speedbooten. Tenslotte zal gebeuren waarvoor men voorlopig in de Golf alleen nog maar bang is, men zal oog in oog komen te staan met de strijdkrachten van de Ayatollah.



Rampage
Gebaseerd op één van de populairste arcade-hal spellen van het afgelopen jaar is Rampage een spel van rampen en vernietiging in drie dimensies. Er zijn drie verschrikkelijke wezens, die nogal sterk lijken op Godzilla, King Kong en Wolf-man. Deze monsters gaan tekeer tegen wolkenkrabbers, gaan op helicopters en soldaten zitten en eten zelfs onschuldige voorbijgangers op. Al dit verschrikkelijks kan op een Commodore 64 worden gespeeld.

Solid Gold
Deze serie 'klappers' is verder uitgebreid met toppers van Activision, Infocom en Gamestar:
- Hacker II: The Doomsday Papers
- The Transformers: Battle to Save the Earth
- Zork 1
- Hitchhikers Guide to the Galaxy
- Star Rank Boxing/On-Court Tennis

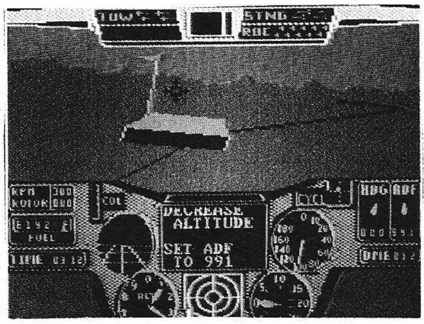
Time Fighter
Deze orgie van strijd en geweld door de eeuwen heen is afkomstig van CRL en beschikbaar op cassette of disk voor de C-64. Beginnend in het stenen tijdperk maakt men de progressie in strijdmethodes en wapens mee van knuppels tot laserkanonnen. Behalve de geschiedenis van het vechten komt ook een mogelijke toekomst aan de orde. Als je al deze methodes hebt gehad dan lijkt het erop dat je in staat moet worden geacht het leukste of interessantste wapen uit de geschiedenis te kiezen, maar nog beter is het als dit spel de werkelijkheid kan vervangen.

Thundercross
Nog een spel van CRL voor de C-64. Een ruimtestofzuiger genaamd Thundercross is van plan het gehele universum op te zuigen. Het is door een groot aantal verdedigings-schepen omringd dus er is bijna niet bij te komne. Toch is het onvermijdelijke doel van dit spel om deze stofzuiger onklaar te maken zodat we niet allemaal in de grote stofzak terecht komen. Natuurlijk rust de redding van het hele universum weer op de schouders van de speler.

Snooker
Speciaal voor de Engelsen en de fans van de BBC-televisie is er een re-release van het snooker spel goedgekeurd door de 'wereldkampioen' Steve Davis. Men kan op diverse manieren snooker spelen, tegen de computer of als demonstratie. Er zijn vele mogelijkheden om de ballen te controleren.

Thunderchopper
Voor de echte oorlogsfanaten en liefhebbers van vlieg-animatie is er van ActionSoft een pakket voor de 64/128 uitgekomen waarin technologie van SubLogic is verwerkt. De speler krijgt de beschikking over een Hughes 530MG Defender helicopter waarmee op diverse niveaus kan worden gevlogen, van eenvoudige landingsoefeningen tot de meest gevaarlijke gevechtsmissies.

Cinemaware
Een begrip dat staat voor interactieve spelsoftware of interactieve film en werd bedacht door Bob Jacob, de president van Cinemaware corporation. Na de succesvolle titels Defender of the Crown, SDI, The King of Chicago en Sindbad and the Trone of the Falcon, komen er twee nieuwe titels uit. The Three Stooges is een comedy classic waarin de van het witte doek bekende figuren optreden. Ze proberen een oude dame en haar drie prachtige dochters te helpen wiens weeshuis wordt bedreigd door de gemene bankier Mr. Fleecem. De stooges proberen geld in te zamelen door prijsvechten, taarten gooien of gekke medische activiteiten. Rocket Ranger is geïnspireerd op klassieke helden als Flash Gordon, Commando Cody en Indiana Jones. In het jaar 1940 is er vanuit de toekomst een raketpak, straalgeweer en geheim decodeerwiel opgestuurd om te vorkomen dat de Nazi's de oorlog winnen. Rocket Ranger is verdeeld in afleveringen, net zoals Hogan's Heroes.



Namaak 1541 drives
Commodore Nederland vermoedt dat er in Nederland op grote schaal namaak 1541 diskdrives aangeboden worden. Commodore heeft van deze diskdrive de speciale chips en het aangepaste operating systeem ontwikkeld en maakt daarom vanaf nu internationaal jacht op de namaak drives, haar producent en aanbieders van de drives. Commodore zal daartoe steekproeven nemen om zo de namaak drives op te sporen.

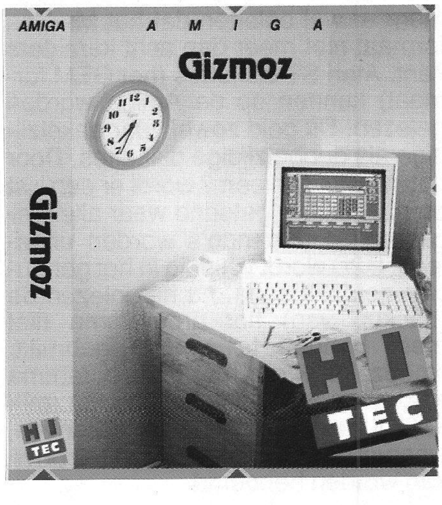
Amiga en audio-video
De Academie voor Schoone Kunsten in Utrecht verzorgt vanaf donderdag 17 maart een aantal driedaagse cursussen over de mogelijkheden van de Amiga voor de Audio-Video producent. Het gebruik van de Amiga in de dagelijkse praktijk staat hierbij centraal. De cursus is bestemd voor iedereen die zich professioneel met audio-visuele communicatie bezighoudt. Er wordt aandacht besteedt aan 2D-animatie, professionele ondertiteling, digitaliseren van videobeelden, 3D 'polygon' animatie en 'ray tracing 3D animatietechniek. Er vinden in 1988 zeven cursussen plaats, waarbij per cursus maximaal 15 deelnemers ingeschreven worden.

Amiga draagtas
Aristos Informatica levert speciale draagtassen voor de Amiga 500 en 1000. De zwarte tassen zijn gepolsterd, de buittenkant is gemaakt van geweven nylon en de boering van niet-krassende nylon/trico. De tassen zijn voorzien van sterke nylon ritsen, een schouder-draagband en sterke handgrepen. De prijs voor de systeemeenheid-tas bedraagt *f* 115,-, de monitortas kost *f* 125,- (excl. BTW). Inl: 03461-3204.

De Ergostick
In Amerika werd in januari op de Consumer Electronics Show in Las Vegas een ieuw model joystick getoond. De Ergostick van WICO Corporation heeft een ergonomisch goed ontwerp, omdat het apparaat speciaal gevormd is om aan te sluiten bij de vorm van de hand. Het apparaat is gedeeltelijk gemaakt van een zacht, op huid gelijkend materiaal, en de vuurknop ligt precies op de juiste plaats voor gebruik met de wijsvinger. In Amerika wordt een prijs van 25 dollar genoemd.

Gizmoz

De Gizmoz Productivity Set is ontwikkeld om volledig alle kwaliteiten van de Amiga te gebruiken door Digital Creations en wordt door Robtek op de markt gebracht. Het programma maakt het gebruik van de computer makkelijker. Het bevat een aantal krachtige gereedschappen om opdrachten makkelijker te kunnen uitvoeren. Op de schijf staat van elk programma een duidelijk uitleg. Helaas is deze uitleg, evenals de uitgebreide handleiding in het Engels geschreven. Het programma wordt geleverd op een onbeveiligde disk, zodat de routines probleemloos naar een harddisk kunnen worden geschreven wat voor vele (serieuze)gebruikers een uitkomst is.
Memopad is een klein, in het geheugen blijvend file waarin teksten kunnen worden opgeslagen. Bijvoorbeeld aantekeningen die tussendoor even bekeken kunnen worden.
Calendar is een in het geheugen blijvende kalender die het mogelijk maakt om tussendoor een datum op te zoeken tussen 1900 en 2099.
Rollodex biedt u de mogelijkheid om namen en adressen op te slaan en deze snel te kunnen opzoeken. Er kunnen kaarten worden opgeslagen van 5 rijen van elk 36 karakters.
Black book is het centrale opmaak programma voor de eerder genoemde programma's. Printer layouts kunnen hiermee worden gemaakt.
Hotkey geeft de mogelijkheid aan een bepaalde toets een definitie te koppelen. Deze instellingen kunnen ook gesaved worden.

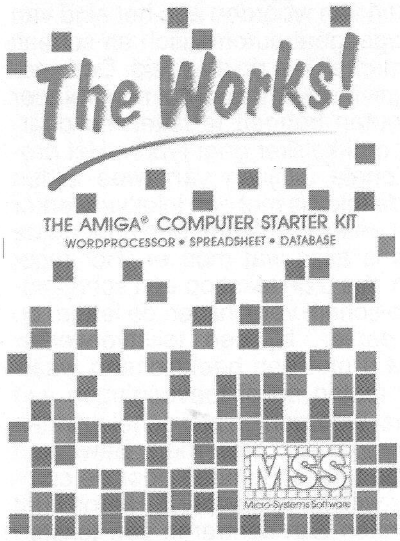


Gizmoz

Scientific Calculator geeft de Amiga een extra dimensie als wiskundige rekenmachine. Ook printen behoort tot de mogelijkheden.
Financial Calculator geeft u de beschikking over een rekenmachine die speciaal is toegespitst op de financiele wereld.
Programmer Calculator Deze rekenmachine heeft vele mogelijkheden die door programmeurs met open armen zal worden ontvangen.
Fastprefs is een utility die het mogelijk maakt instellingen van de preference snel te veranderen.
QuickMouse is een in het geheugen blijvend programma dat het werken met de muis sneller maakt zonder dat de nauwkeurigheid verloren gaat.
Announce is een handig hulpmiddel bij het experimenteren met de Amiga en zijn verschillende kwaliteiten.
SuperLife computers zijn er niet alleen om te werken maar ook spelen moet een van de mogelijkheden zijn is de basis van deze optie.
Graph is een eenvoudig en flexibel hulpmiddel om grafische afbeeldingen op het scherm te toveren.
Cuckoo Clock is een stukje gereedschap om de tijd zeer realistisch op het scherm weer te geven.
IconMaker hiermee kunnen IFF files tot icon worden omgetoverd.
Terminal maakt het mogelijk Amiga's aan de buitenwereld te koppelen via de 232 interfaces.
PopUp hiermee is het mogelijk een grote variëteit aan informatie als achtergrond scherm te geven
Compress maakt files korter. Bij data-overdracht is een kortere tijd nodig om de informatie over te brengen
Encrypt maakt van een file, voor een ander, een onleesbare tekst.
SetPriority hiermee maakt u de Amiga duidelijk welke opdrachten voorrang hebben bij het uitvoeren van de multitasking.

THE WORKS

Nieuw in Nederland is het geïntrigeerde software pakket the Works. Het is een Spreadsheet, database en wordprocessor in één. Deze software is op één disk bijelkaar gebracht zodat het steeds wisselen bij het werken tot het verleden behoort. Dit is zeker voor diegene die veel wisselen tussen database, spreadsheet en de tekstverwerker een grote tijds en ergernis besparing. Het maakt het werken een stuk eenvoudiger doordat er met dezelfde pull down menu's wordt gewerkt voor de verschillende modules waaruit het programma is opgebouwd. Dit houdt ook in dat er voor de verschillende programma's niet steeds een nieuwe handleiding behoeft te worden bestudeerd.
Als spreadsheet is gekozen voor **Analyze**. Een spreadsheet is niets anders



The Works

dan een uitgebreide rekenmatrix. Hiermee kunnen uitgebreide rekenkundige bewerkingen snel en foutloos worden uitgevoerd. Het bestaat uit een aantal verticale en horizontale rijen. Verticaal zijn deze gekenmerkt door een letter en horizontaal door een cijfer. Ieder vakje dat verkregen wordt door de kruising van de rijen en kolommen wordt een cel genoemd. Iedere cel heeft hierdoor een unieke code gekregen bv. A-1, G-12, k-5 enz. In iedere cel kan een nummer, een label of een formule worden opgeslagen. De grote sterkte van een spreadsheet wordt hiermee aangetoond. Tellen, bewerken, vergelijken alles is zowel horizontaal als verticaal mogelijk. Veranderd er ergens één of meerdere gegevens dan is een correctie zo uitgevoerd, alleen de waarde of formule in de desbetreffende cel veranderen en de gehele berekening is bijgewerkt. Hierdoor is het bijvoorbeeld mogelijk een bedrijfsresultaat steeds up to date te houden. Gegevens die zijn ingevoerd kunnen eenvoudig in 3 dimensionale grafieken op het beeldscherm worden getoond. Voor de ervaren gebruiker is er een speciale macro taal om de gegevens nog sneller te kunnen invoeren. Een extra pluspunt is de compatibiliteit met het bekende software pakket Lotus 1-2-3.

Scribble is de uitgebreide tekstverwerker die in het geheel is opgenomen, hierin zit alles wat er van een moderne tekstverwerker mag worden verwacht. Het intypen van tekst is erg eenvoudig. Men hoeft aan het einde van een regel niet zelf naar het begin terug te gaan, dit verzorgt het tekstverwerking pakket zelf. Het afbreken van woorden aan het eind van de regel gaat automatisch en scheelt aanmerkelijk in de snelheid. De tweede tijdwinst komt doordat men minder op fouten behoeft te letten en daardoor makkelijker gaat typen. Het professionele uitlijnen van twee zijden van de tekst is met een tekstverwerker geen enkel probleem, een druk op de toets is alles wat men er voor moet doen. Als u de tekst op een schrijfmachine schrijft verschijnen de letters op het papier, bij een tekstverwerker werkt u met een beeldscherm waardoor fouten en of toevoegingen van letters, woorden en zelf gehele zinnen kinderspel zijn geworden. Bewerken van de teksten kunnen door de computer worden uitgevoerd. Hierbij kunt u denken aan centreren van teksten het opnieuw indelen van teksten of in een andere letter afdrukken van (stukken) van de tekst. Bij een aantal tekstverwerkingspakketten is het zelf mogelijk dat er op de tekst bewerkingen worden uitgevoerd. Kolommen kunnen worden geteld, BTW-percentages automatisch berekend. Dit alles houdt wel in dat er aan een tekstverwerkings pakket hoge eisen moeten worden gesteld. Scribble heeft naast deze mogelijkheden ook een spellingskontrole en een mailmerge mogelijkheid. Er kunnen tot 4 documenten tegelijk op het scherm worden gebruikt.

Organize als laatste maar zeker niet als slechtste is de database die het geheel complimenteert. Een database is eigenlijk niets anders dan een verzameling gegevens. hierop zijn een groot aantal bewerkingen mogelijk. Een database wordt verdeeld is een elektronische kaartenbak. Hierin zijn de kaarten de records, en hierop staan een aantal velden, dit zijn de gegevens. Op elk record en de velden is het mogelijk te sorteren. Ook bestaat de mogelijkheid een voorwaarde mee te geven bv. alles dat groter is dan .. De uitkomsten zijn zowel naar het beeldscherm als ook naar de printer te sturen. Eén van de sterke punten van Organize zijn de Project icons, automatisch wordt een database geopend en worden de gegevens bewerkt. Rapporten zijn volledig aan uw eigen wensen aan te passen en kunnen worden gesaved en geprint. Gegevens kunnen worden geblokkeerd zodat onbevoegden nooit inzage in deze gegevens kunnen krijgen.

Gegevens vanuit de drie onderdelen kunnen van de ene naar de andere module worden overgezet. Het pakket maakt gebruik van de unieke grafische en kleur mogelijkheden van de Amiga. Wat heeft u nodig om met dit uitgebreide pakket te kunnen werken: Een Commodore Amiga met minstens 512K met één of twee diskdrives, een Kickstart 1.1 of 1.2 en natuurlijk het geïntegreerde pakket the Works.
De leverancier is Cat & Korsh (010-4507696) en het pakket kost nu ter introduktie slechts *f* 499,--.

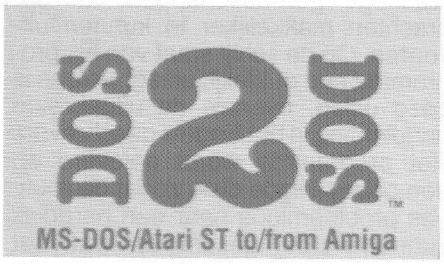
KICKSTART 2

Dol op motertrails en in het bezit van een Amiga dan is dit je kans, allereerst ga je met een motor over allerlei hindernissen zo kun je je kunsten tonen en zo min mogelijk brokken makend aan de finish komen. Bij de kleinste vergissing maak je een mooie duikvlucht, en met een grandioze salto over je motor heen gelukkig zonder verdere brokken krijg je de kans om weer verder te gaan. Er zijn diverse hindernissen te nemen zo hoef je niet raar te kijken als je over telefooncellen heen moet springen, bewegende pijpen moet omzeilen, en nog veel meer van dat schoons. Bij daglicht valt dit niet mee maar in de nacht zijn deze hindernissen extra moeilijk Maar de leukste mogelijkheid is toch wel om je eigen hindernisbaan te ontwerpen. Hierdoor is het mogelijk de meest onmogelijke hindernissen te maken. Nu kan je zelf de moeilijksgraad bepalen. Maak het niet te moeilijk want je vergist je snel in je kwaliteiten als trailcoureur.
Kickstart 2 is uitgebracht door Mastertronic (prijs onbekend).



Kickstart

DOS 2 DOS

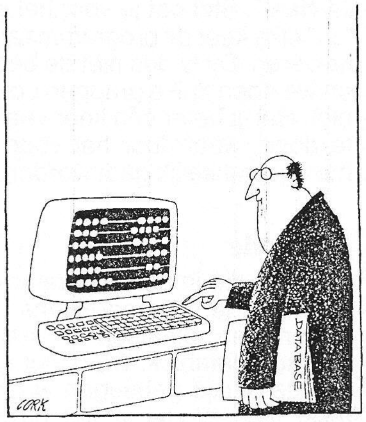


Dos to Dos zet AmigaDos over naar Ms-Dos en Pc-Dos en omgekeerd. Dit biedt ideale mogelijkheden voor teksten die op de Amiga zijn geschreven over te zetten naat Ms-Dos en dan met de professionele desktop pakketten te bewerken. Zelf overzetten naar een zetmachine behoort nu tot de mogelijkheden. Dos 2 Dos kan alle files van AmigaDos, ja zelfs van de Atari ST overzetten naar Ms-Dos. Programma files van Ms-Dos zoals .exe en .com files worden wel overgezet maar kunnen op de 68000 microprocessor van de Amiga niet werken. Het zelfde geldt voor het omgekeerde Amiga programma's werken niet op een IBM of een IBM kloon. D2D werkt alleen vanuit de Command Line Interpreter of kortweg het CLI. Het programma maakt geen gebruik van de Intuition interface, windows en de muis. Het programma leest en formateerd 5.25 en 3.5 inch diskettes onder Ms-Dos. Wildcards zijn bij het invoeren van de filenamen toegestaan maar er wel opletten dat het Ms-Dos formaat niet meer dan acht karakters kent. Atari ST diskettes (het GEM formaat) kunnen op de Amiga worden gelezen. Dit geld zowel voor enkelzijdige als dubbelzijdige diskettes. Door het intypen van een ? wordt er een lijst op het scherm getoond waar alle toegestane commando's worden uitgelegd. D2D wordt volledig in het geheugen gelezen zodat na het inlezen de (backup van) het programma niet meer behoeft te worden gebruikt. Noodzakelijk om met dit programma te kunnen werken is een Amiga (min 256K) met minstens twee diskdrives, waarvan er dan één als Ms-Dos drive kan worden benoemd.

In deze elfde aflevering van de cursus machinetaal gaat Tjipke van der Land verder met het uitwerken van de assembler-mogelijkheden. In dit deel komen met name een aantal instructies aan de orde die wat dieper worden besproken dan in het vorige deel.

Machinetaal

door Tjipke van der Land



Deel 11: Assemblerinstruktie

De vorige keer heb je kunnen zien hoe de assembler werkte met daarbij nog enkele instructies waarmee te experimenteren valt. Deze keer gaan we wat dieper in op de nieuwe instructies, voor diegenen die het de vorige keer niet zo snel konden volgen.

Het belangrijkste tot nu toe is dat je ongeveer begrepen hebt hoe de instructie's van de assembler werken. De vorige keer heb je kunnen zien hoe we een ASCII string ergens uit het geheugen hebben gehaald en vervolgens in het video geheugen hebben neer gezet, met de geïndexeerde x methode.

tabel 1.

X REGISTER	ADRESOPGAVE	ECHTE ADRES
$0	$300	$300
$1	$300	$301
$2	$300	$302
$3	$300	$303
$4	$300	$304
$5	$300	$305

Geïndexeerde x-methode

Het grote verschil met de manier die we eerst hebben uitgevoerd was dat je veel minder instructies nodig had, maar ook dat de lengte van de ASCII string niet meer zo belangrijk is. Hij blijft namelijk net zo lang karakters ophalen totdat hij de waarde nul tegen komt. Eerder stopt hij gewoon niet. In dit geval kun je natuurlijk nooit meer dan 256 karakters ophalen, omdat het x register geen waarde kan bevatten die hoger is dan 256 vanwege zijn acht bits breedte. Met een paar truukjes kan dat natuurlijk wel, maar niet volgens de regels van dit programma. Het is overigens heel gebruikelijk om aan het eind van een string een stopcode te plaatsen wat dus niet altijd een nul hoeft te zijn. Het kan dus ook hexadecimaal FF zijn. Op deze manier kun je een heleboel tekst die normaal in een programma voorkomt ergens achtereenvolgend in het geheugen neerzetten, alleen gescheiden door de stopcode. Op deze manier hoef je alleen maar het juiste adres van de string op te geven, en voor de rest wordt het wel afgewerkt.
Voor het uitvoeren hiervan is natuurlijk wel een programma nodig, want zoals wij weten gaat dit niet automatisch.
Het programma van de vorige uitgave is bijvoorbeeld wel geschikt te maken om zoiets uit te voeren. Daarbij moeten wij ons wel realiseren dat dit programma bij elk woord dat geprint moet worden aanwezig moet zijn. Dat dit te veel ruimte in beslag neemt zal wel duidelijk zijn. Dit probleem is op te lossen door hiervan een subroutine te maken. Dus voor elk woord wat wij willen gaan printen zetten we wat basisgegevens in een register of geheugenlokatie voordat we de routine aanroepen. Deze gegevens zijn bijvoorbeeld, op welk adres moet het eerste karakter opgehaald worden, en waar moet het geprint moet worden.
Als je dit soort gegevens zo neerzet dat de subroutine weet waar ze staan dan hoef je dit programma maar één keer te maken en bespaar je weer een heleboel kostbare geheugenbyte's.

Subroutine

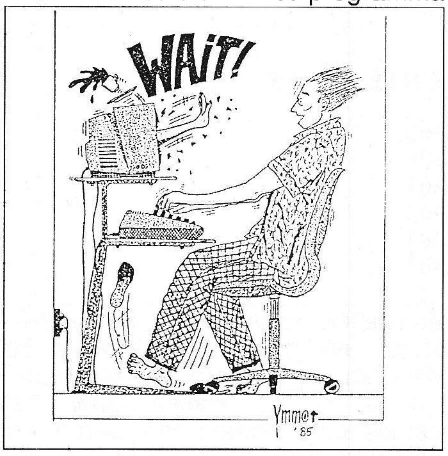
Voor het realiseren van deze subroutine is weer veel infomatie nodig en we moeten het programma van de vorige keer drastisch wijzigen zodat we de parameters ergens kunnen ophalen uit het geheugen of register. Deze kennis zullen we vanwege de complexe samenstelling in twee keer behandelen.
We willen dus door middel van een routine meerdere teksten op verschillend plaatsen op het scherm laten zien. We gaan er vanuit dat we beschikken over een databestand, dus een plek waar we alle teksten hebben staan. We moeten goed weten op welk adres elk woord of tekst begint. Dit is op te lossen door het geïndexeerd indirekt adresseren. Dit proces is niet echt eenvoudig te begrijpen, maar met een beetje geduld kom je hier gegarandeerd uit. Voor de duidelijkheid leggen we nog een keer uit wat geïndexeerd was en daarna indirekt. Het geïndexeerd laden van de accu in ons voorbeeld ging door een adres aan te wijzen, waar bij de waarde van het adres de waarde van het x of het y register werd opgeteld. Stel dat de waarde in het x register $5 is en het adres is $300. Dan wordt de waarde voor de accumulator geladen van adres $300 + $5 dus $305.

In het tabelletje boven kun je zien dat de geadresseerde waarde waar de accu z'n getal kan laden steeds de optelling is van de waarde in het x register + de waarde die je opgeeft. Houd er wel rekening mee dat de waarde in het x register niet uit zichzelf verandert. Je moet dus zelf de waarde verhogen of verlagen, met de instructies INX (verhoog de waarde in x met 1) of DEX (verlaag de waarde in x met 1). Doe je dat niet dan wordt de accu steeds met hetzelfde getal geladen, en had je hem dus beter absoluut kunnen adresseren. In het programma van de vorig uitgave kan je dit terug

vinden. Van dit programma willen we dus een subroutine maken, zodat we dit programma niet steeds hoeven tussen te voegen bij elk minuscuul woordje dat we willen laten zien. Bijvoorbeeld in een programma " Printen Ja/Nee ". Stel dat je voor het woordje "Ja" elke keer dit programmaatje moet invoeren. Dit is dus niet de bedoeling, en we doen zulke grappen ook beslist niet. Het is beter één keer veel moeite te doen, waarrdoor het voor de rest heel gemakkelijk gaat worden.

Stopcode

De tekst die in een programma voorkomt kun je gewoon tijdens het programmeren plaatsen in het zogenaamde datablok. Dit is het gereserveerde stukje geheugen waar je alle tekst neerzet die in het programma



staat. Hier begin je gewoon met het eerste woord wat je gaat gebruiken, met daarachter het stopcode. Bij het volgende woord ga je gewoon door achter de stopcode van het eerste woord. Je moet nu wel het beginadres van het tweede woord onthouden of opschrijven. Dit is raadzaam omdat je straks nog moet weten op welk adres bijvoorbeeld woord 15 staat. Persoonlijk vind ik het een beetje pittig om 15 hexadecimale getallen tijdens het programmeren te onthouden. Dus opschrijven!! In het onderstaande voorbeeld krijg je een indruk hoe de teksten in het geheugen komen te staan (zie tabel 2.)

Hier moet je dus weten dat woord 1 is te vinden op adres $300 en woord 2 op adres $312, want de eerste string van letters zijn 17 stuks dus $11 + de stopcode is tezamen $12. Dit opgeteld bij $300 is $312. Het adres van het eventuele derde woord zal dus $312 + $F +$1 = $322 zijn. Waar we naar toe willen is straks voor het springen naar de subroutine een register te laden met een waarde welke wijst naar het woord wat je wil laten afdrukken op het scherm. Stel dat je string 20 bijvoorbeeld "Printen Ja/Nee" wil afdrukken op het scherm, dan laad je bijvoorbeeld een register of geheugenlocatie met het getal $14. Door middel van het geïndexeerd indirekt adresseren weet de 65xx precies het woord te vinden.

tabel 2.

. 0300 Voer de Naam in :
(stopcode)Voer Adres in :(stopcode)
| |
Begin adres $300
Begin adres $312
van eerste woord
van tweede woord

Indirekt

Deze truuk ligt zoals je waarschijnlijk al vermoedt bij het magische woord indirekt. Wat is nu indirekt ?? Dit is namelijk hetzelfde als bij ons in het dagelijkse leven bijvoorbeeld via een tussenpersoon bij de goede persoon terecht komen, dus indirekt. Dit is al eens eerder uitgelegd in een eerdere uitgave, maar om het geheugen weer wat op te frissen zullen we het nog een keer beknopt doornemen. We nemen even als voorbeeld dat we met z'n allen aan het ballen gooien zijn, en we heten voor het gemak even Jan, Piet en Hans. Hans vraagt aan Piet "gooi mij die bal eens even toe!" Vervolgens gooit Piet de bal naar Hans. Stel even dat deze bal de Data is, dan wordt deze bal/data rechtstreeks van Piet naar Hans gestuurd. Hier spreken we dus over een direkte overdracht, of absoluut. Stel dat Piet de bal niet had maar Jan. Dan roept Piet tegen Jan van gooi die bal even naar Hans, die hem vervolgens naar Hans gooit. Dan heeft Hans de bal/data indirekt gekregen. Wat moeilijker gesteld Hans heeft Piet geadresseerd, om data te sturen, maar krijgt deze data uiteindelijk van Jan via Piet. Op deze manier werkt het in de computer ook, alleen niet met die mooie namen, maar met kale nullen en enen.

Nulpagina

Voor het aanwijzen van het adres waar de woorden staan krijgen we deze keer ook te maken met de zogenaamde nul pagina. Deze kreten zijn in het begin al eens aan de orde geweest, als je het niet meer weet is dat geen ramp, want we herhalen nog even in het kort wat ze betekenen.
De nulpagina is het gedeelte in het geheugen, waar het highbyte $00 is. Om data te halen van de nulpagina hoef je maar te adresseren met 1 byte, dus geen 2 byte zoals bij het absoluut adresseren. Hierdoor wordt een programma zowel sneller als korter. Wat we straks doen is voorbeelden plaatsen van indirekt adresseren. Verwacht dus niet dat de wens van een volledig subroutine in vervulling gaat, want we gaan hier stapje voor stapje naar toe werken. Het eerste stapje dat we gaan doen is indirekt geïndexeerd adresseren met het Y register. Er zijn namelijk twee manieren van adresseren, met het X register en met het Y register. We beginnen met het Y register omdat dit het dichtst het voorbeeld van de vorige keer benadert. Het indirekt adresseren zonder gebruik van registers is niet mogelijk.

Bij de instructie LDA in de kolom indirekt staat een opcode B2 voor het indirekt adresseren. Deze instructie werkt niet, want de 65xx herkent deze opcode niet als een instructie. Schrik dus niet als bij het experimenteren met deze instructie de 65xx vastloopt.

Er is een heel groot verschil tussen het in dit geval adresseren met het X register of het Y register. Het verschil is het beste duidelijk te maken met een paar programma's als voorbeeld, waarvan we achteraf weer het verhaal gaan doornemen wat hierboven staat. In de voorbeelden maken we het principe van de verschillende manieren van adresseren duidelijk. Op deze manier zal het je snel duidelijk zijn hoe het gaat werken.

In ons voorbeeld gaan we een paar karakters uit het geheugen halen en ergens anders weer plaatsen. De karakters die we ophalen zetten we achter elkaar in het geheugen, en gaan deze op de geïndexeerde indirekte methode aanroepen. Het eerste programma is om drie willekeurige karakters die achter elkaar staan, op te halen en weer te plaatsen in het geheugen. Hier kun je met de monitor de op te halen getallen op de goede plek zetten, door in te voeren .m 1500 1550 <return>. Dan op het juiste adres de juiste getallen invoeren. (tabel 3.)

Zoals je ziet staan deze drie getallen vanaf adres $1500. In ons tweede voorbeeld gaan we deze zelfde getallen ergens willekeurig uit het geheugen halen, en weer achter elkaar opslaan, en dit met behulp van het X register wat anders werkt. De adressen kun je weer invoeren met de monitor of met het hulpprogramma. (tabel 4.)

tabel 3.	ADRES	WAARDE
	$1500	$CA
	$1501	$35
	$1502	$9C
	MNEMONIC	**KOMMENTAAR**
.a 1200	LDA #$ CA	laden getal
	STA $ 1500	opslaan getal
	LDA #$ 35	enz.
	STA $ 1501	
	LDA #$ 9C	
	STA $ 1502	
	BRK	terug naar monitor

tabel 4.	ADRES	WAARDE
	$1500	$CA
	$1550	$35
	$1590	$9C
MNEMONIC	**KOMMENTAAR**	
.a 1200	LDA #$ CA	laden getal
	STA $ 1500	opslaan getal
	LDA #$ 35	enz.
	STA $ 1550	
	LDA #$ 9C	
	STA $ 1590	
	BRK	terug naar monitor

tabel 5.	ADRES	WAARDE
	$0070	$00
	$0071	$15
	$0072	$50
	$0073	$15
	$0074	$90
	$0075	$15
	MNEMONIC	**KOMMENTAAR**
.a 1200	LDA #$00	laden getal
	STA $70	opslaan getal
	LDA #$50	enz
	STA $72	
	LDA #$90	
	STA $74	
	LDA #$15	1 keer getal laden
	STA $71	drie keer
	STA $73	achtereenvolgend
	STA $75	opslaan
	BRK	terug naar monitor

Tenslotte moeten we nog wat gegevens invullen op de nulpagina, omdat anders het geïndexeerd adresseren via de nulpagina niet werkt. Wat je ziet is dat de adressen van de willekeurig in het geheugen liggende getallen hier achter elkaar komen te staan. Je kan ook deze waarden weer invullen met de monitor, maar ook met het hulpprogramma. (tabel 5.)

Houd er rekening mee dat eerst deze gegevens in het geheugen moeten staan omdat hij anders echt helemaal niets doet, dus niet eerst de programmaatjes intikken en kijken hoe alles verkeerd gaat. Want dat gaat het gegarandeerd. Waar je in het voorbeeld even op moet letten is hoe je de assembler vertelt dat hij met indirekte adressen heeft te doen. Dit doe je door het adres tussen haakjes te zetten. Nu dus verder met ons voorbeeld.

Zoals je ziet lijken de beide programma's sprekend op elkaar, maar vergis je niet, want ze doen heel verschillende dingen. Het is dan ook het beste dat je zelf deze programma's gaat uitproberen om te kijken wat eigenlijk het verschil is. Zoals je uit het programma kan opmaken worden de resultaten vanaf adres $1400 in het geheugen geschreven. Om je niet helemaal in het duister te laten tasten nemen we het verschil tussen de beide instructies onder de loep.

De instructie LDA ($70),Y gaat via het vaste nulpagina adres $70 kijken op welk adres hij z'n waarde moet halen. In dit geval is dat adres $1500 waar hij begint. Het Y register zorgt vervolgens voor de offset vanaf het adres $1500. Dus als het Y register geladen is met $00 dan wordt de waarde van adres $1500 gehaald. Is de waarde in het Y register $01, dan wordt de waarde geladen van adres $1501 enzovoort. Bij de instructie LDA ($70,X) gaat het net andersom. Hier werkt de offset op de nulpagina. Als in dit geval het X-register is geladen met 00, dan gaat hij op de nul pagina op adres $70 + de waarde van het X register ($00) = $70 kijken op wat voor adres hij z'n getal moet halen.

Dus in het kort. In dit geval kijkt de processor in de nul pagina op adres $70 op welk adres hij z'n getal kan vinden. Het adres wat hij leest op $70 op de nul pagina is $1500. Voor het aanwijzen van het volgende adres daar vlak achter moeten wij ons wederom realiseren dat een adres uit 2 byte's bestaat, dus als op adres $70 een volledig 2 byte adres te lezen was betekent dat, dat het eerstvolgende begin van een tweede adres 2 byte verder moet liggen, dus op $72. Je zal in het voorbeeld 2 ook zien dat het X register 2 maal wordt opgehoogd. De reden hiervan zal je nu wel duidelijk zijn. Je kan de beide verschillende instructies goed onderscheiden door een paar kleine details.

Het indirekt adresseren via het Y register is te herkennen door de Y buiten de haakjes, wat duidt op de offset die wordt uitgevoerd op het adres wat hij heeft gelezen op de nulpagina. Het indirekt adresseren via het X register is te herkennen aan de X binnen de haakjes wat duidt op het uitvoeren van de offset op de nulpagina zelf. De waarde's ingevoerd tussen de haakjes betekenen immers voor de assembler dat dit indirekte adressen zijn. Hier laten we het deze keer bij. Succes bij het proberen, en tot de volgende keer, dan gaan we hier weer mee door.

T. v/d L.

INDIREKT GEINDEXEERD ADRESSEREN MET Y REGISTER

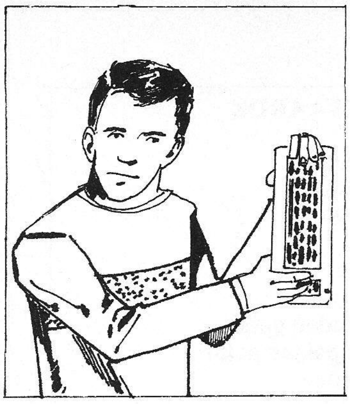
```
           MNEMONIC

.a 1000    LDY #$ 00
           LDX #$ 00
           LDA ($70),Y
           STA $1400,X
           INY
           INX
           CPX #$03
           BNE $(adres lda ($70),y)
           BRK
```

INDIREKT GEINDEXEERD ADRESSEREN MET X REGISTER

```
           MNEMONIC

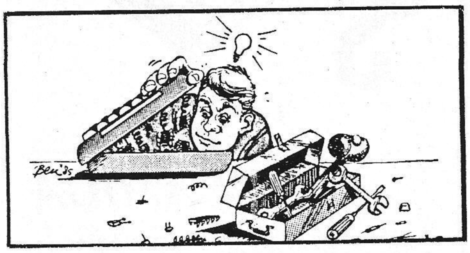
.a 1000    LDX #$ 00
           LDY #$ 00
           LDA ($70,X)
           STA $1400,Y
           INY
           INX
           INX
           CPX #$06
           BNE $(adres lda ($70,x))
           BRK
```



Rene Janssen maakte voor Commodore Info een korte strip. Omdat we dit een leuk idee vonden, drukken we hieronder deze strip op ware grootte af. Niet in de laatste plaats omdat de tekenstijl erg boeiend is.
En waar het belangrijkste artikel deze keer over gaat, dat moet je zelf maar bepalen...



Basic Micro's blijkt populair, want er hebben ons ook dit keer weer heel wat inzendingen bereikt. Mocht het gebeuren, dat die van jou er weer niet bij staat, stuur hem dan nog eens in. Het kan namelijk best zijn, dat er af en toe een in de vergetelheid of de prullebak raakt.



Basic Micro's

Micro's zijn, zo als het woord al zegt, klein. Het gebeurt regelmatig dat we "Micro's" ontvangen die echt te groot zijn om af te drukken, omdat ze een halve pagina zouden opslokken.

Die programma's worden zelden gepubliceerd, omdat de mensen van de Listing-rubriek ook niet van plan zijn om zelf vele vellen vaak handgeschreven tekst over te typen, om vervolgens de fouten te gaan uitzoeken. Beter is het om deze programma's op tape of disk te zetten en dan op te sturen voor de gewone listing rubriek.

Hieronder volgt echter - *bij wijze van experiment* - een programma dat we in twee delen publiceren.

Deze keer dus de eerste helft. Het is een programma dat het Wilhelmus laat horen. Het is geschreven door Herman Mouthaan uit Zwolle en geschikt voor de C-64, maar het moet eventueel met wat wijzigingen ook kunnen lopen op de C-128.

In het programma zaten helaas drie ernstige fouten in de DATA-regels. Dit is dan ook de aanleiding om nog eens te vragen om grote zorgvuldigheid bij het overtypen of overschrijven van ingezonden programma's, een listing is echt beter.

```
10 PRINTCHR$(147)CHR$(5):INPUT"SNELHEID"
   ;SN:SN=SN/3:IFSN<.5ORSN>2THENRUN
20 S=54272:FORL=0TO23:POKES+L,0:NEXT:POK
   ES+5,9:POKES+6,0:POKES+24,15
30 DIMHF(61),LF(61),DR(61):FORP=0TO61:RE
   ADHF(P),LF(P),DR(P):HF=HF+HF(P)
40 LF=LF+LF(P):DR=DR+DR(P):NEXT
50 IFHF<>8440RLF<>84800RDR<>29850THENPRI
   NT"FOUT IN DATA!":PRINTHF,LF,DR:END
60 FORP=0TO61:POKES+1,HF(P):POKES,LF(P):
   POKES+4,33:FORL=1TODR(P)STEPSN:NEXT
70 POKES+4,32:FORT=1TO50:NEXTT,P
80 DATA9,104,500,12,143,500,12,143,500,1
   4,24,250,15,210,250,16,195,250,14,24
90 DATA250,15,210,500,14,24,250,15,210,
   250,16,195,500,15,210,500,14,24,250,12
100 DATA143,250,14,24,500,12,143,1750
```

De rest van de DATA-regels van dit Wilhelmus komt dus de volgende keer aan bod.

SimpStat

Een grafiekje is altijd leuk, dus hier komt weer eens een Micro die er een produceert. Het programma heet SimpStat en is geschreven voor de C-16 en ingestuurd door R. Beijk uit Boxmeer. De in te voeren getallen zijn niet gebonden aan grenzen.

```
10 SCNCLR:INPUT"HOEVEEL GETALLEN";T:DIMA
   (T):L=99999:H=-L:S=0:FORI=1TOT
20 INPUTN:A(I)=N:S=S+N:IFN>HTHENH=N
30 IFN<LTHENL=N
40 NEXT:F=150/H:D=INT(270/(T-1)):G=S/T:
   GRAPHIC2,1
50 DRAW,30,10TO30,150TO300,150TO30,150:F
   ORI=2TOT:DRAW,+D,150TO+0,-3:NEXT
60 DRAW,30,150-INT(A(1)*F):FORI=2TOT:DRA
   W,+0,+0TO+D,150-INT(A(I)*F):NEXT
70 PRINT"GEMIDDELDE ="G:PRINT"HOOGSTE ="
   H" EN LAAGSTE ="L
```

Van dezelfde auteur kregen we zijn visie op het weer van 1987 toegestuurd. De naam van de Micro: druppels. Of zou 'sneeuw' beter zijn?

```
10 SCNCLR:VOL8:B=0:FORI=1TO40:POE4031+I,
   102:NEXT:FORT=1TO960
20 B=3072+INT(RND(1)*40):IFPEEK(B)<>32TH
   EN20
30 POKEB,81:B=B+40:IFPEEK(B)=32THENPOKEB
   -40,32:GOTO30
40 SOUND1,INT(B/4),2:NEXT:GOTO10
```

Scroll

Het volgende programma voor de C-64 kun je maar beter voor het runnen saven als je het wilt bewaren, want daarna kan het niet meer zonder cartridge. Scroll is niet helemaal een juiste benaming, maar beter wisten wij het ook niet. Probeer ook eens de beide of een van beide waardes 238 in regel 50 te veranderen in 206. De checksum wordt dan 7035, respectievelijk 7067.

```
10 T=0:FORA=0TO55:READB:T=T+B:POKE49152+
   A,B:NEXT
20 PRINTCHR$(147):IFT<>7099THENPRINT"DAT
   A ERROR!":END
30 SYS49152:DATA169,44,141,0,128,141,2,1
   28,169,192,141,1,128,141,3,128,169,195
40 DATA141,4,128,169,194,141,5,128,169,2
   05,141,6,128,169,56,141,7,128,169,48
50 DATA141,8,128,76,226,252,238,32,208,2
   38,33,208,234,234,234,76,44,192
```

Split-screen

Micro's voor de C-128 met machinetaal ontvangen we niet zo vaak, vandaar dat we nu de kans grijpen om er twee af te drukken. Ze zijn afkomstig van Bas van Peype

uit Castricum. De eerste heeft een Split-screen tot gevolg, de tweede veroorzaakt bij elke toetsaanslag een ander geluidje.

```
10 C=0:FORT=0TO30:READA:C=C+A:POKE4864+T
   ,A:NEXT:IFC<>3490THENPRINT"FOUT IN
   DATA":END
20 SYS4864:NEW:DATA120,169,13,141,20,3,1
   69,19,141,21,3,88,96,165,216,208,7,169
30 DATA1,133,216,76,101,250,169,0,133,21
   6,76,101,250
```

```
10 C=0:FORT=0TO51:READA:C=C+A:POKE4864+T
   ,A:NEXT:IFC<>5821THENPRINT"FOUT IN
   DATA":END
20 SYS4864:NEW:DATA120,169,24,141,20,3,169
   ,19,141,21,3,169,15,141,24,212
30 DATA141,5,212,141,6,212,88,96,165,213
   ,201,88,208,8,169,0,141,4,212,76,101
40 DATA250,141,0,212,141,1,212,169,33,14
   1,4,212,76,101,250
```

Digitaal

Voor de mensen onder ons die interesse hebben voor hoe computers nou echt rekenen, volgt hier een demonstratie van optellen zonder optellen, gemaakt door Rob Hagelstein uit Doetinchem. Hierbij wordt gebruik gemaakt van het principe van de Full Adder, dat is een optelschakeling die drie bits bij elkaar optelt. Typ overigens in plaats van "^" een "pijltje omhoog". Het bereik van de in te voeren getallen loopt van -32768 to 32767.

```
10 CLR:DIMD(15,2):INPUT"A,B ";A,B
20 FORE=0TO1:FORT=0TO15:D(T,E)=(A/2^T)AN
   D1:NEXT:A=B:NEXT
30 FORT=0TO15:A=D(T,0):B=D(T,1)
40 U=(NOT((NOTA)ANDBAND(NOTC)))AND(NOT(A
   AND(NOTB)AND(NOTC)))
50 U=UAND(NOT((NOTA)AND(NOTB)ANDC))AND(N
   OT(AANDBANDC)):U=NOTU
60 C=NOT((NOT(AANDB))AND(NOT(AANDC))AND(
   NOT(BANDC)))
70 D(T,2)=U:X=X+U*2^T:NEXT:IFU=1THENX=-(
   (NOT(X-65536))+1)
80 PRINT:FORE=0TO2:FORT=15TO0STEP-1:PRIN
   TD(T,E)CHR$(157);:NEXT
90 PRINT:IFE=1THENPRINT" ---------------
   ----------------- +"
100 NEXT: PRINT: PRINT"DE DECIMALE
   UITKOMST IS"X
```

Tekeningen

Sander Jansen uit Raalte zond ons een Micro die het doet op de C- 16, Plus/4 en C-128. Het programma tekent diverse kronkelfiguren. De inzender noemt het zelf Tekenfractals, maar die benaming is niet helemaal correct.

```
10 SCNCLR:GRAPHIC3,1:GRAPHIC1:COLOR0,1:C
   OLOR2,1:COLOR4,1:FORE=0TO650STEP5
20 DRAW1,50,50TO250,E:NEXT:GOSUB60:FORB=
   0TO250STEP5:CIRCLE1,210,B,30:NEXT
30 GOSUB60:FORN=0TO80:BOX1,250,250,N*10,
   100,50:NEXT:GOSUB60:FORX=0TO995STEP3
40 CIRCLE1,200,100,80,X:NEXT:FORA=0TO100
   STEP5:BOX1,100,100,50,50,A:NEXT:GOSUB60
50 FORW=1TO125STEP5:CIRCLE1,150,75,W,30:
   NEXT:END
60 FORY=0TO1500:NEXT:SCNCLR:RETURN
```

Schermflitsers

De routinetjes om flitsende beelden te krijgen lijken ontelbaar.

Bennie Looijaard (12 jaar) uit Eindhoven stuurde een vrij korte flitser in.

```
10 POKE53265,11:FORQ=49152TO49160:READA:
   POKEQ,A:C=C+A:NEXT:IFQ=1190THENSYS49152
20 PRINT"?DATA ERROR":DATA236,32,208,206
   ,32,208,76,0,192
```

Ook onze eigen bijdrage veroorzaakt strepen op het beeld.

```
10 FORT=0TO14:READA:POKE49152+T,A:C=C+A:
   NEXT:IFC=1530THENSYS49152
20 PRINT"?DATA ERROR":DATA120,169,11,141
   ,17,208,78,18,208,46,32,208,76,6,192
```

Printer-lineaal

Wie vaak moet tellen uit hoeveel tekens een geprinte regel bestaat, kan veel baat hebben bij de lineaal die de volgende Micro maakt. Hij werd ons toegestuurd door Harry Eyck uit Neer. Printen, uitknippen en op een kartonnetje plakken en je hebt een prachtige positie-teller. Simpel, maar zeer handig.

```
10 OPEN4,4:CMD4:FORI=1TO8:
   PRINT#4,"0123456789";:NEXT:CLOSE4
```

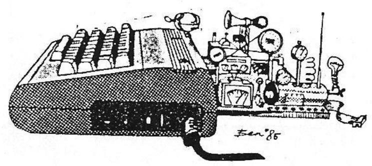
Binaire sprites

Het ontwerpen van sprites kan zeer tijdrovend zijn. Het volgende programma van Sander van der Kooij uit Bleiswijk lost dit op. Het ontwerpen van sprites kan hiermee ook in de DATA-regels gebeuren in plaats van op papier. De routine voor het omzetten van 7-bits binair naar decimaal staat in regel 20.

```
10 V=53248:POKEV+21,1:POKE2040,13:POKEV,
   100:POKEV+1,100:FORN=0TO62
20 READBI$:DE=0:FORB=7TO0STEP-1:DE=DE+VA
   L(MID$(BI$,8-B,1))*2^B:NEXT
30 POKE832+N,DE:NEXT
100 DATA00001100,01100110,00110000
```

etcetera...

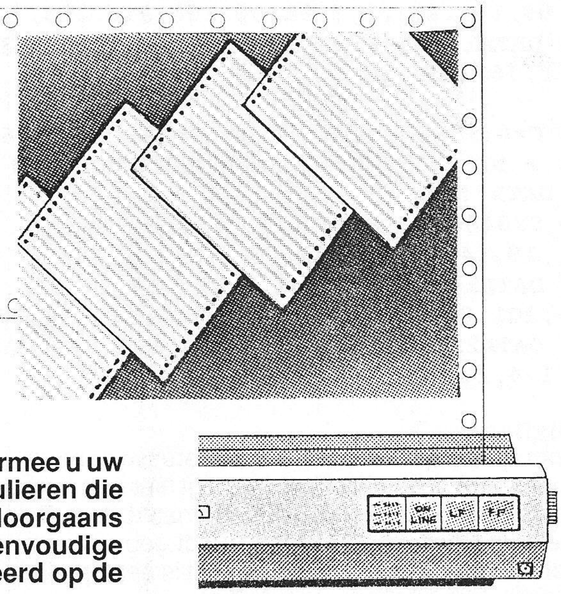
Peter Broekhuizen



**PS. Voor uw inzendingen en alle post:
Postbus 43048, 1009 ZA, Amsterdam.**

De GEOS wereld blijft maar groeien. In minder dan twee jaar is GEOS uitgegroeid tot een complex geheel van professionele, maar vooral ook veelzijdige programma's. De GEOS programmaserie stelt u in staat uw computerervaring verder uit te diepen. En van de nieuwste applicaties uit de GEOS familie is GeoFILE, een geavanceerd database programma.

GEOFILE

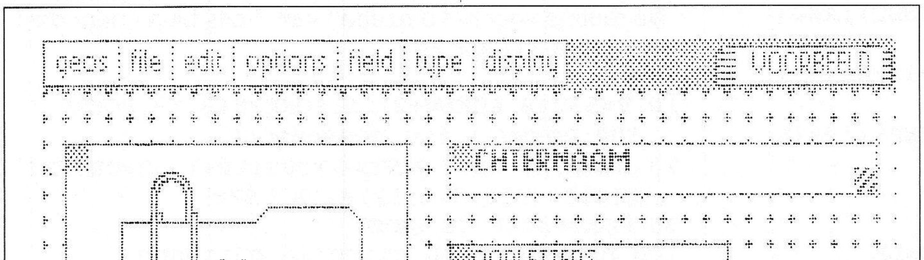


Gegevensbeheer onder GEOS

GeoFILE is het gestandariseerde bestandsbeheer-systeem waarmee u uw gegevens kunt organiseren en opslaan in zogenaamde formulieren die u zelf ontwerpt naar eigen inzicht. Nadat u zo een formulier, die doorgaans worden aangeduid als forms, heeft ontworpen vult u ze op een eenvoudige manier met gegevens. De gegevens worden automatisch gesorteerd op de door u aangegeven rubriek.

GeoFILE biedt vele mogelijkheden om de opgeslagen gegevens op te zoeken en als bruikbare informatie te presenteren, zowel op het scherm als op de printer. Hieronder zal (slechts) een aantal van de vele mogelijkheden welke GeoFILE te bieden heeft nader worden bekeken.

Zoals gezegd maakt GeoFILE gebruik van *forms* (formulieren) om de gegevens in de database (gegevensbestand) op te slaan. Naast een hoofdformulier (*Master form*) met daarin een eigen ontworpen lay-out, bestaat de mogelijkheid om nog eens 15 sub lay-outs binnen de database te ontwerpen. Deze sub lay-outs bevatten gedeelten van het hoofdformulier. Met behulp van deze sub lay-outs kan naderhand een zoekactie of een afdrukspecificatie worden vervaardigd.



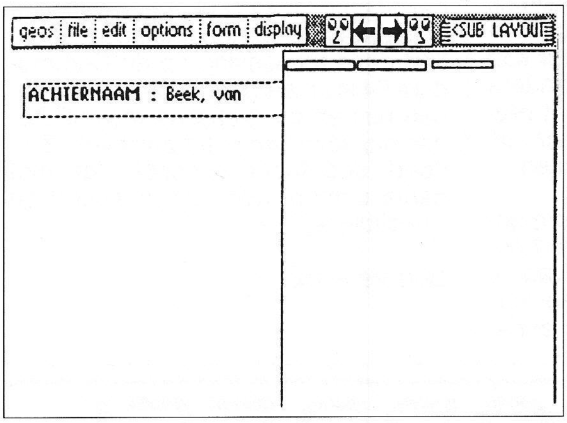
Afgezien van een aantal zeer geavanceerde mogelijkheden, is het werken met GeoFILE een prettige ervaring wanneer men gegevens uit een database wilt uitwisselen met een andere GEOS-applicatie. Zo kan men een mail-merge samenstellen voor gebruik met het tekstverwerkende programma GeoWRITE. Of men maakt een "text-scrap" om in de spreadsheet GeoCALC te laten integreren. Vanuit het tekenprogramma GeoPAINT kan men een database voorzien van plaatjes. Kortom, de combinatie van GeoFILE met andere GEOS applicaties vergroot de bruikbaarheid en veelzijdigheid van GeoFILE.

Het programma wordt het geleverd op diskette met een Engelstalige handleiding. Op de bijgeleverde diskette staan naast de GeoFILE applicatie een nieuwe versie van GeoMERGE. Daarnaast zijn nog een aantal oefenbestanden opgenomen, die u samen met de handleiding kunt doorlopen, teneinde de mogelijkheden van het programma onder de knie te krijgen.

Het GeoFile bestands-ontwerp.

De basis van elk bestand wordt gevormd door de forms (formulieren). Deze forms bevatten de rubrieken waarin de gegevens worden opgeslagen. U ontwerpt deze forms zelf op

een eenvoudige wijze. Nadat de form-pagina op het scherm is verschenen gaat u de rubrieken (fields) in uw bestand definieren. Er kunnen maximaal 64 rubrieken op een form-pagina worden vastgelegd. Elk field wordt benoemd met een field-naam. Zowel de naam als het later in te voeren gegeven worden in een field vastgelegd. Tijdens het ontwerpen van een form kan gebruik gemaakt worden van de mogelijkheid om de fields automatisch te laten richten, zodat een symmetrisch geheel wordt gevormd. Tijdens het ontwerp wordt ook het sort-field gekozen. Dit is de rubriek die GeoFILE gebruikt om de gegevens op te sorteren. Naast het ontwerp van het hiervoor beschreven hoofd-formulier



kunnen maximaal 15 sub lay-outs worden gecreeerd. Deze bevatten geselecteerde rubrieken uit het hoofdformulier.

Voorbeeld

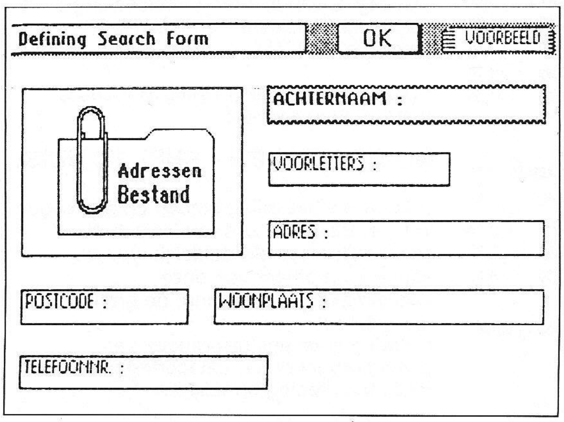
Een sub lay-out kan naderhand worden gebruikt om gedeelten van het hoofd-formulier samen te stellen voor bijvoorbeeld een uitdraai op de printer. Dit kan worden toegelicht met een voorbeeld. Stel u heeft een bestand ontworpen waarin u alle gegevens van uw kennissen heeft opgeslagen. Indien u nu een telefoonlijst of een adreslijst wilt laten printen, hoeft u slechts eenmaal de fields die u daarbij afgedrukt wilt hebben in een sub lay-out te plaatsen. Met behulp van deze sub lay-out worden dan alleen die rubrieken afgedrukt welke u heeft geselecteerd.
Met behulp van de zogenaamde *page preview* kunt u het uiteindelijke resultaat van een form pagina bekijken.

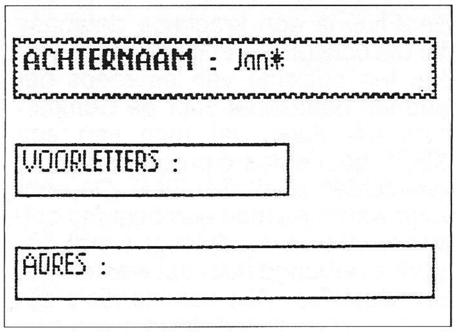
Invoeren en beheren van gegevens.

Door middel van de data entry mode kunt u gegevens in de database invoeren. U wijst met het aanwijspijltje een rubriek (*field*) aan en activeert hem door eenmaal op de muis- of vuurknop van de joystick te klikken. Vervolgens typt u met behulp van het toetsenbord het gegeven in. Een voordeel is, dat als het in te voeren gegeven langer is dan het field, deze toch in zijn totaliteit wordt opgenomen. U kunt de ingevoerde gegevens bekijken met de pijl-iconen, die bovenin het beeldscherm zijn weergegeven. Het invoeren van uitsluitend numerieke gegevens naast een alfanumerieke invoer is ook als voorwaarde in een field op te geven.

Raadplegen van gegevens.

Door een *search-form* (zoekformulier) te definiren kunt u op eenvoudige wijze gegevens op laten zoeken en eventueel wijzigingen aanbrengen. Er wordt bij GeoFILE een onderscheid gemaakt tussen conditioneel zoeken en het raadplegen van gegevens met behulp van zogenaamde "wild cards". Het conditioneel zoeken is verreweg de eenvoudigste methode. Hierbij vult u in een rubriek de conditie in waarop gezocht moet worden. U laat vervolgens door GeoFILE zoeken, door een van de gezichts-iconen te kiezen.
Bij het conditioneel zoeken kan gebruik gemaakt worden van condities als groter dan, kleiner dan, groter/kleiner of gelijk aan of ongelijk aan. Het definieren van combinaties met deze condities behoort eveneens tot de mogelijkheden.
Om het zoeken in een bestand enigszins te beperken bestaat de mogelijkheid om gebruik te maken van zgn. "wild cards". Hiervoor worden het **?** (vraagteken) en het * (asteriks) gebruikt. U wilt bijvoorbeeld iemand in uw bestand opzoeken die Jansen heet. U weet echter niet meer zeker of





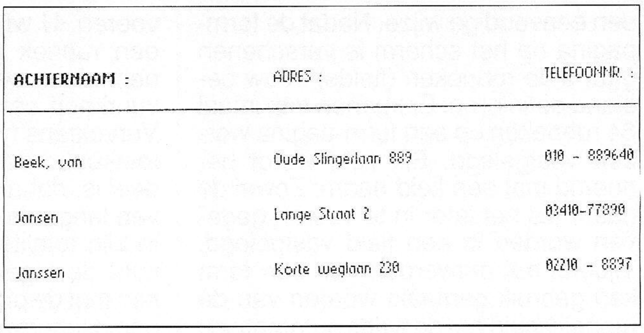
de naam Jansen met een of twee letters S werd geschreven. U voert dan in de te zoeken rubriek de naam Jans* in. Het programma zal na het starten van de zoek-actie alle Jansens met zowel n S of met twee letters S tonen. Of u zoekt naar een postcode, maar u weet de letters niet meer. U voert dan bijvoorbeeld 1781 ?? in. Het programma zal dan niet alleen de door u gezochte postcode vinden, maar ook gegevens als 1781 33 en andere postcodes die beginnen met 1781.
Met behulp van deze methoden kunnen zeer complexe zoekspecificaties worden gedefinieerd. Het is zelfs mogelijk om in een zoekspecificatie niet alleen de zoekargumenten te definiren, maar ook op te geven welke rubrieken dan automatisch gewijzigd moeten worden. Ook het bijwerken (updaten) van opgeslagen informatie kan naast de handmatige methode, automatisch door GeoFILE worden gedaan, door middel van de zoek- en vervangfunctie.

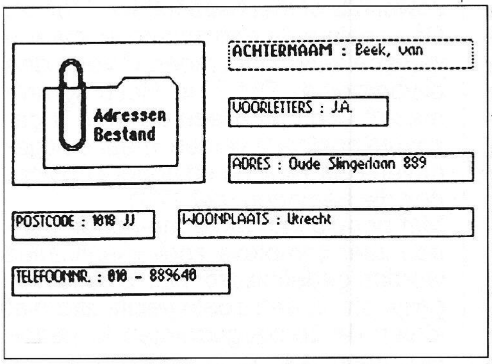
Printmogelijkheden

Uiteindelijk zult u lijsten willen produceren om uw actuele gegevens bij de hand te hebben. GeoFILE biedt hierin vele mogelijkheden en variaties. Zo bestaat de mogelijkheid om meerdere formulieren op een vel papier geprint te krijgen. Daarnaast kunt u kiezen uit etiketten, indexkaarten of gewoon uitvoer op kettingpapier. Ook het laten printen in kolommen behoort tot de mogelijkheden.

Wat u op uw afdruk wilt hebben beslist u zelf met behulp van de sub lay-outs die u eerder had samengesteld. Verder biedt de print-mode de mogelijkheid om rubrieksnamen af te drukken of weg te laten, en of de rubrieks-omlijning al dan niet moet worden afgedrukt. U kunt uiteindelijk één, meerdere of alle opgeslagen formulieren laten printen. Kortom u beschikt over een volwaardig database-programma.

GeoFILE is een krachtige database die als concurrerend gezien kan worden ten opzichte van de reeds bestaande databases voor de Commodore 64. Alles wat men van een GEOS geörienteerd programma mag verwachten biedt GeoFILE. De eenvoud waarmee men een bestand ontwerpt, vult en raadpleegt wordt natuurlijk verhoogd door het werken met een DeskTop. Alle commando's zijn te selecteren uit pull-down menu's en sub-menu's. Daarnaast is het gebruik van windows en iconen op een Commodore 64 op zich al een prettige ervaring. GeoFILE draait standaard op een CBM 64 of een Commodore 128 in de 64-mode. Berkeley Softworks heeft echter al meegedeeld dat de echte 128-versie van GeoFILE binnenkort leverbaar is.





Conclusie

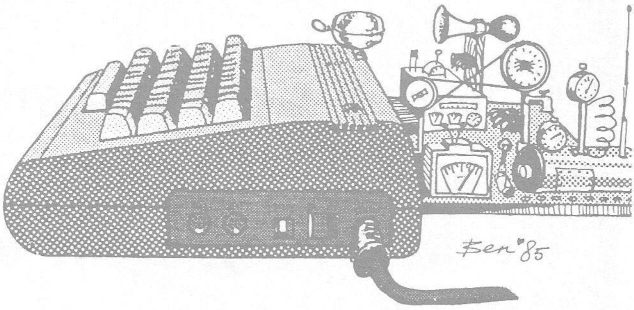
Er kan worden gesteld dat GeoFILE een flexibel en eenvoudig te leren pakket is. Het kost niet veel moeite om de vele gebruiksmogelijkheden in een relatief korte tijd meester te worden. Enig begrip betreffende gegevensbeheer is echter aan te bevelen, maar dat geldt natuurlijk ook voor ander geavanceerde databasepakketten.
Naast al deze mogelijkheden valt ook de snelheid waarmee het programma werkt op. Als enige nadeel kan worden opgemerkt dat als u andere sorteer-rubrieken selecteert het programma enige tijd nodig heeft om de gegevens anders te rangschikken.

Met GeoFILE bent u in staat om gegevens van allerlei soort bij te houden, op een wijze die u alleen van de veel duurdere computers mag verwachten. En in combinatie met andere GEOS geörinteerde programma's bent u in staat om omvangrijke rapporten te produceren.
De prijs van GeoFILE zal in Nederland waarschijnlijk rond de *f* 100,- komen te liggen, waarmee ook al weer blijkt dat dit programma zeker een gunstige prijs/prestatie verhouding heeft.
Met deze toepassing op de Commodore 64 is nu een ware opmars begonnen binnen de grafische mogelijkheden die deze computer ons biedt. Een goed doordacht concept dat met name uitmunt door zijn eenvoud en veelzijdigheid.

Bert Venema

Het geheel van deze (al) 21 lessen Basis Basic vormt samen een goede basis om met het programmeren van de Commodore in Basic uit de voeten te kunnen. Jan Bodzinga behandelt stap voor stap de juiste weg van het programmeren.
Deze keer zullen we samen met hem eens kijken hoe we op een aardige manier data in onze software kunnen verwerken. En dan bedoelen we niet direct computerdata maar eerder hebben we kalenderdata op het oog.

Basis Basic



deel 21: Van datum naar data

Het omgaan met computersoftware, ook op de schaal waarmee met de Commodore wordt gewerkt, is in de zakelijke omgeving voor een groot deel tijdsafhankelijk. Dat betekent dat voor veel bewerkingen van de computer het erg belangrijk is te weten op welke dag van het jaar de gegevens worden ingevoerd of verwerkt. De algoritmes, die voor het werken met datumgegevens zijn uitgedacht liegen er dan ook niet om. Het grote probleem schuilt daarbij in het feit, dat juist de dagen van het jaar, de maanden en de tellingen per dag, week en maand niet erg gestructureerd zijn. We zullen daarom ook voor (Commodore) Basic in deze aflevering een paar tijdsbesparende datum-routines doornemen.

De dag van het jaar waarin we leven, is voor iedereen van belang. Zonder afspraken hierover is het bijna niet mogelijk een goede samenleveing zoals we die kennen te hanteren en in stand te houden. Zolang het eenvoudige zaken betreft als 'Ik zie je morgen' of 'kom over twee dagen maar terug' dan zal het voor ieder van ons nog wel onthouden zijn; maar zodra we een veelvoud van dergelijke afspraken in ons op moeten nemen laten we het toch vroeg of laat afweten. Daarom is het erg nuttig -en al eeuwenoud- dat we te maken hebben met een kalender.
We zullen niet al te diep ingaan op het fenomeen, dat onze kalender niet is gebaseerd op een tien- of andertallig stelsel. En die reden is natuurlijk niet, dat er in die tijd geen computers bestonden. De natuurlijke beweging van de zon ten opzichte van maan en aarde vroeg eenvoudigweg om een complex en ongestructureerd kalendarium, waardoor we uiteindelijk, met het fors bijstellen van onnauwkeurigheden te maken hebben met een redelijk sluitende tijdrekening.
Het is daarbij aan onze generatie om gegeven de omstandigheden onszelf en ons computertijdperk maar aan te passen aan deze onregelmatige tijdrekening. Veel kundige wetenschappers zijn ons daarbij voorgegaan met het uitdenken van aardige formules, waardoor toch nog enige lijn te ontdekken valt in het rekenen met tijd en datum.

Excursie

Voor we verdergaan met het dateren van computersoftware, lijkt het me leuk een andere onregelmatigheid, waar we dagelijks mee te maken krijgen wat beter te bekijken.
Hoewel we er geen van allen vaak bij stil staan is ook ons geldsysteem niet al te regelmatig. Hoewel gelukkig gebaseerd op het decimale stelsel, kennen we bij het monetaire systeem in ons land toch verschillende grootheden. Vanf de cent - die ook al weer verdwenen is) naar het biljet van 1000 gulden liggen een grote hoeveelheid munten en waardepapieren, waarbij de intervallen tussen de oplopende reeks nogal verschillend zijn. Ook daarmee komen we normaal gesproken bij een computer nogal in de problemen.
Laten we het maar eens op een rijtje zetten, hoeveel verschillende geldseenheden we in Nederland kennen. Als we ook de cent nog mee laten doen, en we beginnen onderaan, dan ziet de tabel er zo uit:

cent	**0.01**
stuiver	**0.05**
dubbeltje	**0.10**
kwartje	**0.25**
gulden	**1.00**
rijksdaalder	**2.50**
vijf gulden	**5.00**
tientje	**10.00**
vijfentwintig	**25.00**
vijftigje	**50.00**
honderdje	**100.00**
vuurtoren	**250.00**
duizendje	**1000.00**

Als je 't optelt een aardige som, en één oogopslag leert, dat het triest gesteld is met de structuur. Waarom geen munt van 50 cent, maar wel van vijftig gulden en weer niet van 500 gulden? Raadsel waar we geen direct antwoord op kennen, maar wel weten dat we ze niet hebben.
Het aardige is nu een computerprogramma te bedenken, dat in staat is, alle bedragen die vanaf het toetsenbord worden ingevoerd om te rekenen naar de minimale hoeveelheid bankpapier en munten dat nodig is om het ingevoerde bedrag samen te stellen.

Geldige Basic

Het werken met geld geeft ons de gelegenheid een aardige routine uit te werken, waarmee we de computer kunnen laten bepalen hoeveel stuks van iedere eenheid in het muntenstelsel in een bepaald bedrag gaan. De basisroutine om dit te kunnen bekijken is in feite helemaal niet moeilijk, je moet er alleen even opkomen.
Het programma dient als eerste te weten hoeveel verschillende geldwaarden we hebben en uiteraard welke

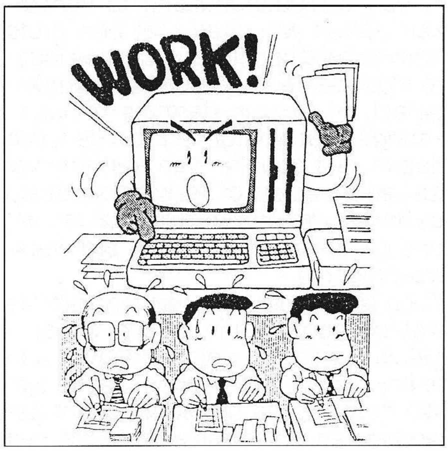
waarde ze stuk voor stuk vertegenwoordigen. De simpelste manier om dat te doen is in dit geval uiteraard een serie DATA opdrachten, te meer omdat de waarde en de hoeveelheid van onze munten en bankpapier niet al te vaak veranderen.
De routine om uit te rekenen hoeveel van ieder in een bepaald bedrag gaan, vereist vooraf enig overleg. Hoe moet dit worden aangpakt.
In een normale situatie zouden we vrij snel te weten komen (zonder computer) hoeveel van ieder in een bedrag gaan. De computer dient het probleem gestructurrd te krijgen voorgeschoteld.
Het geheim zit erin, dat we de DATA opdrachten, die in een numerieke array worden ingelezen, **gesorteerd** van hoog naar laag hebben neergezet. Daardoor vertegenwoordigt het eerste element in de array altijd de hoogste waarde.
Stel we hebben een bedrag van honderdtwaalf gulden, waar we deze som op willen loslaten, zonder computerprogramma, maar wel met de juiste structuur.
We beginnen dan te kijken of de grootste waarde (*f*1000 in het bedrag past. Dat is niet het geval. Ergo nul x 1000. Als tweede is het 250-biljet aan de beurt, en ook dat past niet in de 112 gulden. De derde (*f*100) heeft meer geluk, want daar kunnen we één van kwijt. We hebben dus nu 100 === 1.
Dan houden we nog 12 gulden over om te verdelen, waarbij we verder zonder succes kijken naar de briefjes van 50 en 25 gulden. Na het tientje, die wel kan, houden we nog twee guldens in de pot, die uiteindelijk onder de guldens zullen worden gezet.
Wat gebeurt in werkelijkheid bij deze manier van afchecken?
Als eerste wordt het bedrag **gedeeld** op de mogelijke geldswaarde. Daardoor wordt het aantal bepaald. Vervolgens wordt het bedrag **verminderd** met het aantal X de waarde, waarna het restantbedrag overblijft.
Deze procedure herhaalt zich, totdat het hele bedrag is opgedeeld, en alle mogelijkheden zijn getest. Bekijk het programma maar eens:

```
500 REM DATA GELDWAARDEN
510 DATA 1000,250,100,50,25
520 DATA 10, 5, 2.5, 1.0
530 DATA 0.25, 0.10, 0.05, 0.01
540 DATA -1
```

Als eerste de DATA van geldswaarden. Let erop, dat deze in aflopende volgorde in de regels zijn gezet. Gebeurt dit niet, dan geeft het programma weinig hoop op een juiste berekening. De volgorde is essentieel.

```
10  REM GELD
20  REM
30  REM Omrekenen bedragen
40  REM in geldseenheden.
50  :
60  (c) Commodore-Info / JB
90  :
100 REM initialisatie
110 DIM GD(20) :REM
    GELDWAARDEN
115 DIM TT(20) :REM SOORTEN
120 BD= 0      :REM Bedrag
130 K = 0      :REM Luswaarde
140 TELLER = 1 :REM TELLER
    aantal
```

Tot zover weinig nieuws. De array GD(xx) dient voor het opslaan van de respectievelijke waarde, TT(xx) wordt gebruikt om de hoeveelheid die per bedrag bij een bepaalde waarde hoort (0 of meer) in te bewaren. Standaard staan deze beide arrays op 0.




```
200 REM LEZEN WAARDEN
210 REM
220 READ GD(TELLER)
230 IF GD(TELLER) > 0 THEN
    TELLER = TELLER + 1 :
    GOTO 220
240 TELLER=TELLER-1  REM
    Juiste hoeveelheid
```

In de hiervoorgaande listing wordt als eerste alle waarden van papier en munten ingelezen in de array GD(XX), waarbij de TELLER netjes wordt bijgehouden. Deze variabele is verderop in het programma ook nog van belang, daarom wordt hij in regel 240 op het juiste aantal geld-soorten (13) gezet.
Het vervolg:

```
300 REM BEGIN INVOER BEDRAG
310 PRINT CHR$(147) : REM CLS
320 PRINT:PRINT:PRINT
330 INPUT "Bedrag :";BD

350 REM Berekenen
360 FOR K = 1 TO TELLER
370 G% = INT(BD / GD(K))
380 BD = BD - ( G% * GD(K))
390 TT(K) = TT(K) + G%
400 NEXT K
```

Afgezien van de invoer van bedrag BD, treffen we in de regels 350 tot 400 de kern van dit programma. Het bedrag BD wordt in de loop K steeds vergeleken met één van de waarden uit GD(xx), te beginnen bij de grootste waarde. Om het juiste gehele aantal te bepalen gebruiken we regel 370 een integer-variabele G% en de INTeger functie. In regel 380 verminderen we BD met het aantal (G%) maal de waarde die hier bij hoort (GD(K)), waarna we G% in de arry TT(K) stoppen om het aantal te bewaren.
Simpel en leuk om wat beter te bekijken. Vaak is het juist met deze dingen zo, dat, als je ze eenmaal hebt bekeken en ook werkelijk hebt gesnapt, dan vergeet je een dergelijke eenvoudige manier van programmeren nooit meer en je kunt hem daarna vaak in allerlei varianten toepassen.
Als afsluiting van dit leuke programma printen we natuurlijk de uitkomst nog:

```
440 PRINT
450 FOR K = 1 TO TELLER
460 PRINT
    GD(K);TAB(10);"===";TAB(20)
    ; TT(K)
470 NEXT
499 END
```

Een leuke excursie, waarbij uiteraard nogal wat verfraaiingen het programma nog leuker kunnen maken. Ook kun je van een hele serie bedragen in een overall-lus, gaan uitrekenen hoeveel je van iedere eenheid nodig hebt om een hele fabriek het maandsalaris te laten uitbetalen. Bovendien is het een goeie oefening om de centen uit deze listing te verwijderen, maar dan wel het bedrag af te ronden bij de stuivers, zoals dat tegenwoordig in de supermarkt gebeurt.

Datum

Terug echter naar het onderwerp van dit artikel, de tijdrekening. Vanaf het Romeinse Keizerrijk kennen we in feite al een redelijke jaartelling, waarbij de verstreken tijd van 1 jaar in grote lijnen overeenkomt met de tijd die de aarde nodig heeft om een volledige omloop rond de zon te maken. tijdens die omloop draait de aarde iets vaker dan 366 keer om haar eigen as, waardoor je zou denken dat het jaar minstens 366 dagen moet hebben.
Dat klopt echter niet, want doordat er ook rondom de zon wordt gedraaid, bevindt de zon zich iedere dag op net

een iets andere plek ten opzichte van de voorgaande en navolgende omwenteling. Om een lang verhaal kort te maken, een jaar bestaat uit ongeveer 365,25 dagen. De zaken zijn nog heel wat complexer dan hiervoor omschreven, en de precieze tijd van één jaar ligt op 365,25964 dagen, als we spreken over gemiddelde zonnedagen.
Julius Ceasar was hiervan reeds op de hoogte gesteld door z'n sterrekundigen, waaronde Sosigenes en hij stelde daarom het jaar op 365 dagen en 6 uur, waarbij hij om de vier jaar ook een schrikkeljaar invoerde met daarin 366 dagen. Dit schrikkeljaar maakt het verlies van de 3 maal 6 uur goed van de voorgaande 3 jaren. Op deze wijze kende men tot in de 16e eeuw de jaartelling.
Zo'n 1550 jaar na de invoering van deze Juliaanse kalender bleek toch, dat de onnauwkeurigheid van Julius Ceasar in deze te groot geworden was, want Julius telde een paar minuten per jaar teveel. Maar enkele minuten 1550 keer betekent toch een verschil van meer dan 10 dagen. Dat begon de Paus te vervelen, dus deze Gregorius XIII besloot de hele telling te herzien, en het jaar definitief vast te stellen op 365,2422 dagen, ofwel 365 dagen, 5 uur 48 minuten en precies 51 sec.
Dit had in 1542 tot gevolg dat na 4 oktober ineens 15 oktober voor de deur stond, waarop ineens alle engelse voor de deur van hun bankiers stonden om de vergeten rente op te eisen. Voor de jaartelling betekent deze Gregoriaanse tijdrekening, dat we zelfs nu, meer dan 300 jaar later nog steeds redelijk binnen de telling zitten. Iedere vier jaar een schrikkeljaar, en bij de eeuwwisselingen van 1700, 1800, 1900 **geen** schrikkeljaar. In het jaar 2000 zullen we dus weer **wel** een schrikkeljaar beleven. Wat daarna gebeurt houden we nu maar voor gezien.
Ieder jaar tellen we om precies te zijn 0,00778 dagen teveel. De rest mag je zelfs uitrekenen.

Kalender

Naast de duur van het jaar bestaat onze tijdrekening uiteraard uit meer dingen. Zo hebben we in een jaar meestal 52 weken, soms zelfs 53, zoals in 1987, terwijl ook de 12 maanden nogal voor afwisseling zorgen. Wat dat betreft kunnen we ons veel beter redden met de hebreeuwse tellingen, want daar heeft men geen last van specifieke dagen etcetera. Men telt per week simpel de dagen af als Eerste dag, tweede dag en zo verder.

Het aantal dagen per maand is ook aan niet al teveel regels gebonden, al hebben we wel een standaard gekregen, zodat niet iedereen voor zich maar kan gaan tellen.
De schijngestalten van de maan hebben waarschijnlijk voor de indeling van ons jaar model gestaan. Deze zijn gemakkelijk te bekijken en duren gemiddeld zo'n 29,5 dagen, terwijl een week daar een vierde gedeelte van is. Probleem is echter, dat 12 maal 29,5 beslist minder is dan 365,25. Daarom heeft men indertijd besloten aan enkele maanden in plaats van 30 dagen, 31 dagen toe te kennen, terwijl de tweede maand (februari) als absoluut buitenbeentje 28 c.q. 29 dagen bevat.

Structuur

Het zal voor iedereen wel duidelijk zijn, dat het beslist niet meevalt deze jaartelling op een fatsoenlijke manier in een computerstructuur te verwerken. Ofwel we zitten met een grote hoeveelheid condities, waarmee iedere ingevoerde dagdatum kan worden getest, of we zijn dermate onnauwkeurig in het weergeven van de juiste datum, het bepalen van een interval tussen twee data of het kontroleren op de invoer van een korrekte datum, dat ons computersysteem niet erg werkbaar meer is.
Toch is er - met hulp van wat slimme wetenschappers uit vroeger dagen - gelukkig nog wel enige structuur aan te brengen in het rekenen met de tijd. We hebben ons op dit probleem geworpen, en er zijn een serie leuke, niet al te lange en zeker handige stukjes Basic uit voortgevloeid.
Bedenk wel, dat er in ieder programma wat zich met deze materie bezig houdt een zeker mate van onnauwkeurigheid schuilt. we geloven echter, dat de hieronder gepubliceerde routines probleemloos zakelijk kunnen worden ingezet. Op z'n hoogst verlangen ze eens per honderd jaar een bijstelling, waarvan we de eerste in het jaar 2100 kunnen verwachten. We zijn daarom ook niet aan het programmeren geslagen met de kontrole op het alterneren van schrikkeldagen bij de eeuwwisseling. Bekijk de routines een wat precieser, want ze zijn niet 1-2-3 te begrijpen, al zal in veel gevallen het letterlijk overtypen reeds voldoende zijn.

Basic

Sorry voor de theorie die aan deze listingregels vooraf ging, maar ja, als je zolang bezig bent om uit een volledig ongestructureerde tijdtelling een routine te destilleren, die in een logisch apparaat als een computer aan het werk moet, dan vormt dit type wetenschap op z'n minst een goede ondergrond. Hoe dan ook, we zullen nu echt verdergaan met wat Basic routines.

Als eerste een simpel invoerstuk, waarmee we voor alle programma's standaard een datumstring in een bepaalde variabele zetten:

```
1500 REM INVOER DATUM
1510 REM ALS STRING D$
1520 D$ = ""  :REM INIT
1530 PRINT " DATUM (als
     JJMMDD) ";
1540 INPUT D$
1550 IF LEN (D$) <> 6 THEN
     GOSUB 1600: GOTO 1520
1560 FOR L = 1 TO 6
1570 A$=MID$(D$,L,1 )
1580 IF VAL(A$)= 0 AND A$ <>"0"
     THEN GOSUB 1600: GOTO
     1520
1590 NEXT
1595 RETURN
```

Deze subroutine is in ieder willekeurig programma te gebruiken en doet niet veel anders dan op de plaats van de cursor op het scherm een string vragen, die als JJMMDD de datum moet vormen. Na de invoer worden direct twee basistests uitgevoerd, de eerste in regel 1550, waar wordt bekeken of de ingevoerde string inderdaad precies 6 karakters lang is. Is dit niet het geval, dan wordt via een foutmeldingsroutine (1600 vv) opnieuw een datuminvoer gevraagd.
De tweede test kost iets meer ruimte. Deze loopt van 1560 tot 1590. De lus pakt steeds 1 karakter van D$, via de MID$() functie, zoals die in de vorige aflevering aan de orde is geweest. Van deze 1 letterlange A$ wordt nu bekeken of het inderdaad een digit (cijfer) is, dat door de gebruiker is ingevoerd. Bevat D$ een letter van het alfabet of bijvoorbeeld een leesteken, dan wordt ook naar de foutroutine gesprongen.
Bestaat D$ uit allemaal cijfers, dan volgt geen melding en komt men via RETURN terug in het hoofdgedeelte van het programma.

Melding

Bij een goed programma horen op z'n minst een serie routines, waarin verschillende meldingen zijn ondergebracht. We zagen in de vorige listing al, dat er voor het melden van een foutief ingevoerde datum als twee keer een verwijzing naar subroutine 1600 werd gemaakt. Voor hetzlefde geld hadden we daar de gegevens van de regels 1600-1630 in kunnen voeren,

maar daarmee maken we een onnodig lang (maar wel wat sneller) Basicprogramma.
Hier komen de twee melding-routines:

```
1600 REM DATUM FOUT
1610 REM MELDING
1620 PRINT:PRINT "FOUTE
     DATUM ";D$;" INGEVOERD"
1630 RETURN
1699:
1700 REM DATUM GOED
1710 REM MELDING
1720 PRINT:PRINT "DATUM ";D$;"
     IS KORREKT"
1730 RETURN
```

De manier om deze zo simpele printregels in subroutines af te handelen is zeker de moeite waard. Ook later in het programma kunnen we moeiteloos van dezelfde routines gebruik maken, terwijl we als we een dergelijke melding willen wijzigen, door bijvoorbeeld een andere taal in te voeren, of een kadertje om de tekst te zetten, of we willen alle meldingen per sé op de onderste regel van het scherm, dan hoeven we deze wijzigingen **uitsluitend** in de betreffende subroutine aan te brengen, en niet her en der door het programma heen.

Uitsplitsen

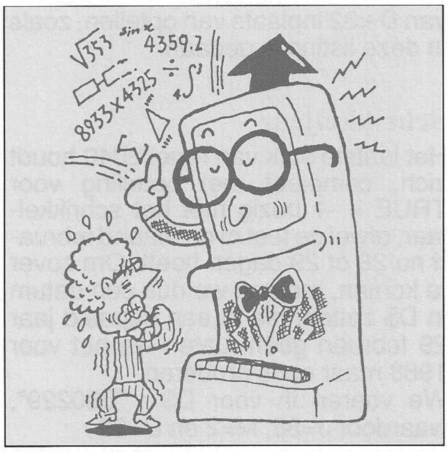
Tot zover hebben we nu een routine om een datumstring (D$) in te voeren, waarvan we met hulp van elementaire tests weten, dat de string precies 6 karakters telt, en dat deze karakters bestaan uit alleen cijfers. Wat we niet weten, is of het formaat van de string goed is, en of uiteraard een geldige datum is ingevoerd. De eerste tests hebben nog niet strings als "110288" en "333333" uitgefilterd. daarvoor moeten we nog een paar regels Basic schrijven.
Als eerste zullen we moeten zorgen, dat de sting D$ op de juiste wijze wordt gesplitst in 3 numerieke variabelen, die voor de duidelijkheid de originele namen J,D,M hebben gekregen. We hebben daar de volgende routine voor:

```
1000 REM DATUMSPLISEN
1010 REM INVOER ALS STRING
     D$
1015 REM D$ = "JJMMDD"
1020 REM WORDT OMGEZET IN
     J,M,S
1030 E = 19 : REM EEUW
     (KONSTANT)
1040 J = 0 : REM JAAR (0-99)
1050 M = 0 : REM MAAND (1-12)
1060 D = 0 : REM DAG (1-31)
1070:
1100 J = VAL ( LEFT$ (D$,2) )
1110 M = VAL ( MID$ (D$,3,2) )
1120 D = VAL ( RIGHT$(D$,2) )
1150 RETURN
```

We zien, dat naast J,M,D ook variabele E met een constante waarde van 19 is ingevoerd. Deze E stelt de eeuw waarin we momenteel leven voor, maar dan gedeeld door 100. Eventueel kan ook E natuurlijk uit een 8 karakterlange D$ worden gehaald. Daarbij moet D$ worden ingevoerd als bijvoorbeeld "19800325". Uiteraard moeten daarvoor ook de andere routines worden aangepast.
Als we tijdens het runnen van deze routine bij regel 1150 zijn aangeland, dan hebben we de beschikking over vier variabelen, die samen de in D$ ingevoerde datum vormen.
Bij invoer van "880412" kennen we als belangrijke variabelen na uitvoer van deze routine dus:

```
D$ = "880412"
E = 19
J = 88
M = 04
D = 12
```

Samen de datum 14 april 1988 vormend.
We gaan verder met de belangrijkste subroutine voor ons eerste datumprogramma.



Maanden

We hebben al de nodige theorie over de maanden van het jaar achter de rug, en op de kleuterschool werden de namen ons al ingestampt. Om de volgende routine wat beter te begrijpen, zullen we echter toch nog even wat beter met het aantal dagen per maand vertrouwd moeten raken.
Als ezelsbrug telden we vroeger de knokkels van een hand, beginnend bij januari. Iedere knokkel stelde een maand voor met 31 dagen, ieder dal een maand met ofwel 30 of 28/29 dagen. Daarmee krijgen we het volgende schema:

1	**januari**	**31**	**2**	**februari**	**28/29**
3	**maart**	**31**	**4**	**april**	**30**
5	**mei**	**31**	**6**	**juni**	**30**
7	**juli**	**31**			
8	**augustus**	**31**	**9**	**september**	**30**
10	**oktober**	**31**	**11**	**november**	**30**
12	**december**	**31**			

Kort samengevat hebben de maanden met nummer 1, 3, 5, 7, 8, 10 en 12 een duur van 31 dagen.
Maand 2 heeft er 28/29 afhankelijk van het jaar en de maanden 4, 6, 9 en 11 hebben 30 dagen.

Testroutine

Met deze lichte structuur van maanden en dagen in ons hoofd kunnen we de volgende routine best aan. Deze routine vormt het hert van het eerste programma en hier wordt bekeken of een juiste datum in D$ in ingetoetst:

```
2000 REM TEST GELDIGE DATUM
2010 REM DATUM IN J,M,D
2020 :
2030 X = -1 AND M > 0 AND M < 13
     AND D > 0 : REM GOEDE
     INTEGERS
2040 X = X AND D < 32 +(M=4 OR
     M=6 OR M=9 OR M=11) +
     (M=2)*(3+(INT(J/4)*4=J))
2050 IF X = 0 THEN GOSUB 1600
     : REM FOUTE DATUM
2060 IF X <>0 THEN GOSUB 1700
     : REM GOEDE DATUM
2080 RETURN
```

De sleutel van deze moeilijke routine is de vlag X. Als X een waarde heeft anders dan 0, X=TRUE ofwel X ongelijk 0, dan hebben we te maken met een geldige datum. We zien in de laatste regels (2050 en verder) dat daar deze test wordt uitgevoerd en naar gelang de waarde van X naar routine GOED (1700) of FOUT (1600) wordt gesprongen. Ook na RETURN kan natuurlijk in het hoofdprogramma op grond van de waarde van vlag X nog altijd worden bekeken of deze test geslaagd is.
Regel 2030 en 2040 zullen wel wat meer problemen geven. Toch wil ik het er niet bij laten zitten, en aan de hand van een bespreking per regel toch wat licht op deze Boolse vergelijkingstests laten schijnen.
We beginnen bij regel 2030:

```
2030 X = -1 AND M > 0 AND M < 13
     AND D > 0
```

Als eerste zien we, dat aan X de waarde -1 wordt gegeven. Deze waarde zet X op TRUE (ongelijk 0). Daarna kijken we of voor M=maand en D=dag een waarde is ingevoerd binnen de grenzen.

We kunnen deze ene regel ook beschrijven met de volgende lijst van opdrachten:

```
10   REM DUMMYTEST
20   X = -1
30   IF M < 0 THEN X=0
40   IF M > 13 THEN X=0
50   IF D < 0 THEN X=0
```

In deze vier regels gebeurt exact hetzelfde als in regel 2030 van onze routine. De wijze van 2030 geeft een snellere manier en schrijfwijze weer. Er wordt gebruik gemaakt van de AND functie in Basic, waarbij als uitkomst altijd een waarde ongelijk aan nul (FALSE) gegeven wordt als de vergelijking fout is. Anders is de uitkomst logisch TRUE. We hebben in regel 2030 alle vergelijkingen gewoon naast elkaar gezet en als ook maar één van de tests niet goud zou zijn, wordt de waarde voor X meteen al nul, ofwel FALSE.

Regel 2040 borduurt tamelijk complex verder op de waarde die X heeft gekregen uit regel 2030. Maar nu worden met name de dagen in de specifiek maand getest:

```
2040 X = X AND D < 32 +(M=4 OR
     M=6 OR M=9 OR M=11) +
     (M=2)*(3+(INT(J/4)*4=J))
```

Deze regel is in een paar stukken uit elkaar te plukken.
Als eerste krijgt X de waarde uit de vorige vergelijking meer (TRUE of FALSE). Daarnaast wordt getest of de dagen (D) kleiner is dan 32. De test op kleiner dan 0 hebben we voor reeds in regel 2030 uitgevoerd.
Als D groter is als 31 krijgen we voor dit gedeelte van de regel een FALSE. Vervolgens komt bij dit gedeelte de test of de maanden misschien horen in de categorie van maanden met 30 dagen, ofwel de nummers 4,6,9 en 11, zoals we eerder hebben gezien.
Stel maar even, dat we hebben ingevoerd "880412". M is dan 4 en D=12. Deze test loopt ot hiertoe met de volgende waarden:

```
X= -1 AND D< { 32 + ([M=4] -1
OR 0 OR 0 OR 0 ) }
```

ofwel

```
X = -1 AND -1  --> X = -1
```

Een moeilijk te begrijpen stuk, zeker als je weinig weet van het gebruik van AND en OR. Het programma is er hier op gebaseerd, dat als TRUE een waarde van -1 terug komt. Door deze waarde (-1) voor de maanden met 30 dagen, op te tellen bij de vergelijking D < 32 krijgen we dus D < 32 + (-1) ofwel D < 31, waardoor we precies bereiken wat we willen, namelijk de test of D voor deze waarden van M (4,6,9,11) kleiner is dan 31. Simpel, maar kom er maar op en weet later bij het debuggen nog maar eens hoe het nu precies zat.

Ongelijkheid

Er zijn nogal wat verschillen in de interpretatie van TRUE en FALSE en de weergave daarvan per computer. De meeste machines hanteren gelukkig 0 voor FALSE en -1 voor TRUE. Er zijn echter computer, die een 1 als waarde voor TRUE kennen. Het is erg gemakkelijk, om dat bij je eigen Commodore (of andere computer) te proberen, door het volgende programmaatje te runnen:

10 Commodore (of andere computer) te proberen, door het volgende programmaatje te runnen:

```
10   REM TRUE-WAARDE
20   M=4
30   PRINT "FALSE = "; (M=2)
40   PRINT "TRUE  = "; (M=4)
```

Aan de hand van deze uitkomsten kun je het programma voor je eigen computer aanpassen. Want als je voor TRUE een waarde 1 krijgt, moet je de uitkomst van M natuurlijk aftrekken van D <32 inplaats van optellen, zoals in deze listing is gedaan.

Schrikkeljaar

Het laatste stuk van regel 2040 houdt zich, compleet met optelling voor TRUE = -1 bezig met het schrikkeljaar, ofwel de test of de maand februari nu 28 of 29 dagen heeft. Om zover te komen, moeten we dus een datum in D$ zetten, die in een bepaald jaar 29 februari geeft. Laten we het voor 1988 maar eens proberen.
We voeren in voor D$: "880229", waardoor J=88, M=2 en D=29
De test van regel 2030 geeft voor X een waarde -1.
Voor regel 2040 ziet het plaatje er als volgt uit (zet je maar even schrap:

```
X= -1 AND D< { 32 + (0 OR 0
OR 0 OR 0 ) }
```

tot hier dus geen bijtelling, maar M=2 en daar komen we in de rest van de regel aan toe:

```
(M=2) * (3 + (INT(J/4)*4=J))
```

ingevuld krijgen we daarvoor:

```
(-1) * (3 + (INT(88/4)*4=88))
(-1) * (3 + (22)*4 =88 ))
(-1) * (3 + (88 = 88))
(-1) * (3 + (-1))
(-3 + (-1) = -2)
```

waardoor de vergelijking in regel 2040 voor een schrikkeljaar als 1988 wordt:

```
X= -1 AND D< { 32 + (0 OR 0
OR 0 OR 0 ) } + -2
X= -1 AND D< 32-2 --> D <30
```

De invoer van 29/02 is dus geldig en dat klopt want 1988 is een schrikkeljaar.
Vullen we bijvoorbeeld in het hiervoorgaande als jaar 1987 in, dan krijgen we uiteindelijk als uitkomst :

```
X= -1 AND D< { 32 + (0 OR 0
OR 0 OR 0 ) } + -3
X= -1 AND D< 32 -3 --> D <29
```

Pittig, maar bijzonder efficiënt. Bekijk het goed en vul zonodig per groep wat waarden in voor M en D, waardoor alles naar ik hopen mag wat duidelijker wordt. Ik hoop in ieder geval, dat deze moeilijke regel goed wordt gedrukt, want het missen van één groter- of kleiner-dan teken of een haakje zal alles in de war schoppen op een zodanige manier, dat je er geen touw meer aan vast kunt knopen.

Als laatste voor deze aflevering het hoofdprogramma, waarmee alles wat we tot hier toe met betrekking tot de datum hebben gehad aan elkaar kan worden gebreid:

```
10   REM Datum controle
20   REM Test op geldige datum
100  REM Begin programma
110  PRINT CHR$(147)  : REM CLS
120  PRINT : PRINT
130  GOSUB 1500  : REM DATUM
     INVOER
140  GOSUB 1000  : REM
     UITSPLITSEN
150  GOSUB 2000  : REM TESTEN
     DATUM
999 END
```

Dit gedeelte is erg kort, mede doordat het gehele programma in feite niet meer is dan een subroutine. Na het schoonmaken van 't scherm wordt niet veel meer gedaan dan de verschillende routines achterelkaar uitvoeren, waarna het programma afgelopen is. Vanuit de routine om de werkelijke test uit tevoeren wordt op grond van de waarde van X al een melding gegeven, zodat dit in het hoofdgedeelte niet hoeft te gebeuren.

Slot

De volgende keer zullen we met de tijdrekening en het dateren wat verder puzzelen en programmeren. De materie is er leuk genoeg voor en bovendien zullen routines vna dit type zeker niet misstaan in de meest uitvoerige commerciële programma's.
Sterkte

Jan Bodzinga